Stateless Near Optimal Flow Control with Poly-logarithmic Convergence

Baruch Awerbuch[1]* and Rohit Khandekar[2]

[1] Johns Hopkins University. baruch@cs.jhu.edu
[2] IBM T.J.Watson Research Center. rkhandekar@gmail.com

Abstract. We design completely local, stateless, and self-stabilizing flow control mechanism to be executed by "greedy" agents associated with individual flow paths. Our mechanism is very natural and can be described in a single line:

> If a path has many "congested" edges, decrease the flow on the path by a small multiplicative factor, otherwise increase its flow by a small multiplicative factor.

The mechanism does not require any initialization or coordination between the agents. We show that starting from an *arbitrary* feasible flow, the mechanism always maintains feasibility and reaches, after poly-logarithmic number of rounds, a $1+\epsilon$ approximation of the maximum throughput multicommodity flow. Moreover, the total number of rounds in which the solution is *not* $1+\epsilon$ approximate is also poly-logarithmic. Previous distributed solutions in our model either required a state since they used a primal-dual approach or had very slow (polynomial) convergence.

1 Introduction

The goal of this paper is to optimize resource allocation in a decentralized network architecture where different applications compete for shared network resources in a "greedy" manner, without explicit coordination with each other, while being subjected to some regulatory constraints that limit their behavior.

In this paper, we focus on the Flow Control version of the multi-commodity flow problem in a distributed environment. The essence of Flow Control is to decide how much flow of a commodity is *admitted* (rest is *rejected*), assuming infinite flow demand for each commodity, and assuming routing is pre-determined to go over a single path. Flow control is used by TCP and is considered a classical problem in the theory of networking, with numerous articles dedicated to this topic.

The flow control problem is a variant of the multicommodity flow problem in a directed capacitated graph, with a collection of *commodities*, each characterized by the following: source (where the flow is originated), sink (where the flow ends up), benefit (the utility of this flow), and a fixed path from source and the sink that must be used by this commodity. The collection of all the flows must satisfy capacity constraints, namely the total flow on each edge cannot exceed its capacity.

* Partially supported by NSF grants CCF 0515080, ANIR-0240551, CCR-0311795, and CNS-0617883.

Intuitively, it is clear that the flow control needs to be accomplished by requiring congested paths to decrease their flows and non-congested paths to increase their flows. However, there is an inherent instability problem, in this concurrent decision making environment — some paths may fluctuate between being congested or non-congested.

Stateless algorithms. "Statelessness" is often a desirable feature of an optimization in a distributed environment [3]. In a stateless and "local" solution, one desires that the flows make their routing decisions in a cooperative but uncoordinated manner, without having access to a global clock and without being able to properly initialize and synchronize their individual executions. The flows only observe the current network congestion, without being able to pin-point the individual contributions of other commodities, and without keeping any memory about the past. As pointed out in [3], statelessness implies a number of other features desirable in networks and distributed systems with unreliable components. *Self-stabilization:* it is a classical notion in the theory of robust distributed systems [8, 13, 9, 6, 5] means that the solution can withstand adversarial sequence of "hard reset" events, namely, crashes accompanied with loss of all memory contents. *Incremental and local adjustment:* Even if changes occur in the network topology or demand pattern, the algorithm does not need to be restarted. Rather, the algorithm adjusts the flows in a local and incremental manner, without disrupting the flows that are not affected. *No global clock:* Algorithms should not be driven by a global clock; however we assume that local clocks generate perfectly synchronized (but un-numbered) *rounds*, without maintaining a global time.

2 Greedy Stateless Maximum Benefit flow framework and our results

Consider a directed capacitated graph $G = (V, E, c)$ where $c : E \to R^+$ is the capacity assignment to edges. We consider the set $\mathcal{P}$ of commodities, each associated with a path p along which it flows and benefit b_p per unit of flow. We identify the commodity with its path p. Let $f(p)$ denote the flow of commodity p. For each edge e, total flow on e is $f(e) = \sum_{p:e\in p} f(p)$. Given a flow f, the *load* of an edge, maximum network load, and total network benefit (flow value) are respectively

$$\ell_f(e) = \frac{f(e)}{c(e)}, \qquad |\ell_f| = \max_{e\in E} \ell_f(e), \qquad \text{and} \qquad U(f) = \sum_{p\in\mathcal{P}} b_p \cdot f(p).$$

The objective is to compute a flow assignment f to commodities satisfying the capacity constraint: $|\ell_f| \le 1$, and *maximizing* the throughput $U(f)$.

Greedy framework. In this paper, we focus on a specific framework for stateless flow control, that we call a "greedy" framework. We imagine an "agent" associated with each commodity p. The agent has benefit b_p associated with sending per unit flow. The restrictions on the agents are as follows:

- *Penalty:* The network imposes at all times a cost on an agent p determined by a "penalty" function $cost(p) = \sum_{e\in p} \phi'_e(\ell_f(e)) = \sum_{p:e\in p} \phi'_e(f(p)/c(e))$ where $\phi_e(\ell_f)$ is a certain function of the congestion on edge e, and ϕ'_e is its derivative.

- *Inertia:* if $cost(p)$ is within $1 \pm \alpha$ factor of its benefit b_p, then the agent p cannot change its flow $f(p)$,
- *Speed limit:* the agent p can only modify (increase or decrease) a β fraction of its flow, namely can change its flow by $(1 \pm \beta)$.

Assuming that the agents act *greedily* and *selfishly* and try to maximize their profit, an agent p will maximally increase the flow if it is *significantly profitable*, namely $cost(p) < (1-\alpha)b_p$, and will maximally decrease the flow if it is *significantly unprofitable*, i.e., $cost(p) > (1+\alpha)b_p$. Our flow control mechanism specifies cost function ϕ', the inertia parameter α and speed limit β.

Theorem 1. *Our mechanism that is presented in Fig. 2 guarantees that starting from an* arbitrary *feasible flow, the flow always remains feasible, and reaches a* $1+\epsilon$ *approximate solution in time upper-bounded by*

$$\tilde{O}\left(\frac{\log k \cdot \log^7(mCB)}{\epsilon^6}\right).$$

Moreover, the total number of rounds in which the solution is not $1+\epsilon$ *approximate is also* $\tilde{O}\left(\frac{\log k \cdot \log^{O(1)}(mCB)}{\epsilon^{O(1)}}\right)$.

Here k and m denote the number of commodities and edges in the network respectively, C is the ratio of maximum to minimum edge capacity, and B is the ratio of maximum to minimum benefit. $\tilde{O}$ hides $\log(\frac{\ln m}{\epsilon})$ factors. We remark that we have not attempted to get the best possible powers of $\log m$ and ϵ in the convergence time. The emphasis is on a stateless distributed solution with poly-logarithmic convergence.

2.1 Previous work

Stateful algorithms. In centralized or distributed setting, efficient "primal-dual" algorithms in the stateful model have been widely studied for network flows [11, 10, 18, 4, 2]. Most of these algorithms share features like exponential dual variables with our algorithms. However, these algorithms crucially depend on maintaining a state, e.g., proper initialization or some global information about the current solution, and perform globally optimum updates in each round. Many of these algorithms initialize the flows to zero. Thus they have to be restarted whenever the instance changes due to change in the network or commodities; and do not satisfy the *incremental* and *local adjustment* property. The packing/covering LP algorithm of Plotkin-Shmoys-Tardos [17] or the multi-commodity flow algorithm of Awerbuch and Khandekar [3] fall short of being stateless since they have to keep track of the maximum violation in a constraint or the global maximum congestion of the current solution. The algorithm of [3] converges in time *linear* in the maximum path-length.

Stateless algorithms. Garg and Young [12] presented a stateless flow control algorithm. While their algorithm resembles ours in the case of flow control, the convergence time of their algorithm depends *linearly* on the ratio B of the maximum and minimum benefit of the flows. This linear dependence is inherent to their algorithm due to a severe

Stateless	Distributed	Problem [citation]	Convergence time
no	no	packing/covering LP [17, 11, 10, 18, 14]	$m \cdot [\log(m)/\epsilon]^{O(1)}$
no	yes	packing/covering LP [16, 7, 18, 15]	$[\log(mn)/\epsilon]^{O(1)}$
no	yes	multi-comm. flow routing [4, 3]	$L \cdot [\log(m \cdot C)/\epsilon]^{O(1)}$
yes	yes	multi-comm. flow control [12]	$B \cdot [\log(C)/\epsilon]^{O(1)}$
yes	yes	bipartite load balancing [1]	$[\log(m)/\epsilon]^{O(1)}$
yes	yes	multi-comm. flow control [*this paper*]	$[\log(m \cdot CB)/\epsilon]^{O(1)}$

Fig. 1: A comparison of some combinatorial $(1+\epsilon)$-approximation algorithms for multicommodity flows. Here n denotes the number of variables and L denotes the maximum-path-length.

limit on how much flow of a commodity can increase in a single round. Their algorithm is based on the packet drop-rates at various routers/links. Recently, Awerbuch-Azar-Khandekar [1] presented a stateless algorithm for a special case of load balancing in bipartite graphs. Their algorithm and techniques, which do not use any exponential duals, do not appear to generalize to arbitrary LPs and hence new techniques are required.

3 The mechanism

By scaling, we assume without loss of generality, that $\max_p b_p = 1$, $\min_p b_p = 1/B$, $\min_e c(e) = 1$, and $\max_e c(e) = C$. The algorithm to be executed by an agent controlling the flow on path p is given in Figure 2. The main procedure executed by each agent is given in Figure 2a. It calls procedures ROUTEMETRIC, CONTROL, FLOWUPDATE in each round. In the rest of the paper, we use round and time interchangeably.

The ROUTEMETRIC **procedure.** Procedure ROUTEMETRIC in Figure 2b sets the most basic parameters. Let $f(e) = \sum_{p:e \in p} f(p)$ be the current flow and $\ell_f(e) = \frac{f(e)}{c(e)}$ be the congestion of an edge e. We define $\Phi_e(f(e)) = c(e) \cdot \exp[\mu(\frac{f(e)}{c(e)} - 1)]$ to be a "penalty" function where $\mu = \epsilon^{-1} \ln(mCB)$ and its derivative $\Phi'_e(f(e)) = \mu \cdot \exp[\mu(\frac{f(e)}{c(e)} - 1)]$ to be the "cost"of edge e. The cost of a path is simply the total cost of the edges on that path.

The CONTROL **procedure.** Procedure CONTROL in Fig. 2c sets the relevant parameters of flow control, namely of the amount of flow that one decreases or increases on a link.

We choose $\alpha = \frac{1}{4\mu}$. Intuitively, this is the accuracy of our cost comparisons, costs within $1+\alpha$ factor will be considered essentially equal. Note that if the flow of a certain commodity over an edge is small (e.g., zero), the multiplicative increase by factor $1+\beta$ is ineffective. To bootstrap the increase in the flow, we allow an additive increase $\ddot{f}$ in the flow. The parameters β and $\ddot{f}(e)$ are defined so that the cost $\Phi'(e)$ of any edge e changes by a factor of at most $\frac{\alpha}{4}$ in any round. Note that the flow $f(e)$ increases in one round by at most $\beta f(e) + k\ddot{f}(e)$. Here k denotes the number of agents or paths $|\mathcal{P}|$. Note that in the procedure FLOWUPDATE, the flow on e is increased only when $f(e) < c(e)$. Thus to ensure that the cost of e does not increase by more than a $1 + \frac{\alpha}{4}$

```
Repeat in each round:
  1. Input (read) $f(p)$ and $f(e)$ for each $e \in p$
  2. Call RouteMetric, Control, FlowUpdate
```

(a) Procedure Main for commodity p.

```
Procedure RouteMetric
  1. Let $\mu \leftarrow \epsilon^{-1} \ln(mCB)$.
  2. Define $\Phi_e(f(e)) = c(e) \cdot \exp[\mu(\frac{f(e)}{c(e)} - 1)]$ for all $e \in p$
  3. Define $\Phi'(p) = \sum_{e \in p} \Phi'_e(f(e)) = \sum_{e \in p} \mu \cdot \exp[\mu(\frac{f(e)}{c(e)} - 1)]$
```

(b) Procedure RouteMetric for commodity p.

```
Procedure Control
  1. Define parameters: $\alpha \leftarrow \frac{1}{4\mu}$ ; $\beta \leftarrow \Theta(\alpha \cdot \frac{1}{\mu})$ and $\ddot{f}(e) \leftarrow \frac{1}{\mu} \cdot \frac{c(e)}{k} \cdot \ln(1 + \frac{\alpha}{8})$
```

(c) Procedure Control for commodity p.

```
Procedure FlowUpdate
  1. if $\Phi'(p) < b_p \cdot (1 - \alpha)$, then $f(p) \leftarrow \max\{f(p)(1 + \beta), \min_{e \in p} \ddot{f}(e)\}$.
  2. if $\Phi'(p) > b_p \cdot (1 + \alpha)$, then $f(p) \leftarrow f(p)(1 - \beta)$.
```

(d) Procedure FlowUpdate for commodity p.

Fig. 2: The maximum benefit flow mechanism

factor, it is enough to make sure that $\exp[\mu(\beta c(e) + k\ddot{f}(e))/c(e)] \leq 1 + \frac{\alpha}{4}$. We in fact set β and $\ddot{f}(e)$ so that $\exp[\mu \cdot \beta] \leq (1 + \frac{\alpha}{4})/(1 + \frac{\alpha}{8})$ and $\exp[\mu \cdot (k\ddot{f}(e)/c(e))] \leq 1 + \frac{\alpha}{8}$.

We set $\beta = \Theta(\alpha \cdot \frac{1}{\mu})$. We set $\ddot{f}(e) = \frac{1}{\mu} \cdot \frac{c(e)}{k} \cdot \ln(1 + \frac{\alpha}{8})$. For all k commodities, the total additive increase in the flow is at most $c(e) \cdot \frac{1}{\mu} \cdot \ln(1 + \frac{\alpha}{8})$ yielding a multiplicative increase in the cost of at most $1 + \frac{\alpha}{8}$.

The FlowUpdate **procedure.** The crux of our algorithm lies in procedure FlowUpdate (Fig 2d). Each agent p locally compares the cost $\Phi'(p)$ under the current "routing metric" with its benefit b_p. If the cost is significantly lower, i.e., less than $b_p \cdot (1 - \alpha)$, it increases the flow by a factor $1 + \beta$. In case the current flow on p is *very* small, the flow is instead increased *additively* by an amount $\min_{e \in p} \ddot{f}(e)$. If on the other hand, the cost is significantly higher, i.e., more than $b_p \cdot (1 + \alpha)$, the flow is decreased by a factor of $1 - \beta$. If neither of the above conditions hold, the flow on p is kept unchanged.

We note that the initial values of the flows are completely arbitrary, as long as they satisfy the capacity constraints; there is absolutely no coordination between flows in terms of how quickly they act, except that there is at most one action of each flow in each round.

4 Analysis

Notations. Let $U = \sum_{p \in \mathcal{P}} f_p \cdot b_p$ be the overall flow in the network at any given point. Let $\ell_f(e) = f(e)/c(e)$ and $|\ell_f| = \max_e \ell(e)$. Let $\Phi(e) = \Phi_e(f(e)), \Phi'(e) = \Phi'_e(f(e))$, and $\Phi''(e) = \Phi''_e(f(e))$. Let $\Phi(p) = \sum_{e \in p} \Phi(e)$, $\Phi'(p) = \sum_{e \in p} \Phi'(e)$, $\Phi = \sum_{e \in E} \Phi(e)$, and $\Phi' = \sum_{e \in E} \Phi'(e)$. Let $g(p)$ denote the *optimum* flow on path $p \in \mathcal{P}$, let $g(e)$ denote the optimum flow on edge $e \in E$. Let $\Gamma = \sum_{e \in E} f(e) \cdot \Phi'(e) = \sum_{p \in \mathcal{P}} f(p) \cdot \Phi'(p)$ be the cost of the entire flow f. Let $\Lambda = \sum_{e \in E} g(e) \cdot \Phi'(e) = \sum_{p \in \mathcal{P}} g(p) \cdot \Phi'(p)$ be the cost of the optimum flow under the current cost metric. Let $\Gamma(e) = f(e) \cdot \Phi'(e)$ and $\Lambda(e) = g(e) \cdot \Phi'(e)$. Let $\Gamma'(e) = f(e) \cdot \Phi''(e) + \Phi'(e)$. For a path $p \in \mathcal{P}$, $\Lambda'(p) = \sum_{e \in p} \Lambda'(e) = \sum_{e \in p} \Phi''(f(e)) \cdot g(e)$ be the derivative of Λ w.r.t. the flow $f(p)$ on path p. Similarly let $\Psi'(p) = b_p - \sum_{e \in p} \Phi'(e)$ be the derivative of Ψ w.r.t. the flow $f(p)$ on path p.

Definition 1. *We introduce the auxiliary potential function:* $\Psi = U - \Phi$.

Definition 2. *We call the network*

- unsaturated *if* $\Phi' < \frac{1}{B} \cdot (1 - \alpha)$*, i.e., the overall cost of the edges is small.*
- saturated *if* $\Phi' \geq \frac{1}{B} \cdot (1 - \alpha)$*, i.e., the overall cost of the edges is large.*
- cheap *if* $\Gamma < U \cdot (1 - 2\alpha)$*, i.e., the average cost of a path is small.*
- reasonable *if* $U \cdot (1 - 2\alpha) \leq \Gamma \leq U \cdot (1 + 2\alpha)$*, i.e., the average cost of a path is about right.*
- expensive *if* $\Gamma > U \cdot (1 + 2\alpha)$*, i.e., the average cost of a path is large.*

4.1 Preliminary observations

The following lemma states how well a convex function is approximated by its first order linear approximation.

Lemma 1. *For a differentiable convex function* $\Upsilon : \Re^k \rightarrow \Re$*, for any* $x, y \in \Re^k$ *we have*

$$\Upsilon'(x) \cdot (y - x) \leq \Upsilon(y) - \Upsilon(x) \leq \Upsilon'(y) \cdot (y - x)$$

where $\Upsilon'(\cdot)$ *denotes the gradient evaluated at a given point.*

Lemma 2. *The potential* Ψ *does not decrease during the course of the algorithm.*

Proof. The negative potential $-\Psi$ is a differentiable convex function of the flow vector $f = (f(p))_{p \in \mathcal{P}}$. Let f_0 and Ψ_0 (resp. f_1 and Ψ_1) denote the values of f and Ψ in the beginning (resp. in the end) of a round. Using Lemma 1, we conclude that

$$\begin{aligned} \Psi_1 - \Psi_0 = -\Psi_0 - (-\Psi_1) &\geq -\Psi'(f_1) \cdot (f_0 - f_1) \\ &= \sum_{p \in \mathcal{P}} (f_1(p) - f_0(p)) \cdot (b_p - \Phi'_{f_1}(p)) \end{aligned} \quad (1)$$

where ${\Phi'}_{f_1}(p)$ denotes the cost of p under flow f_1. Since the cost of a path does not increase by a factor more than $1 + \frac{\alpha}{4}$ in a single round, we conclude that ${\Phi'}_{f_1}(p) \leq (1 + \frac{\alpha}{4}){\Phi'}_{f_0}(p)$. From the algorithm we conclude that $f_1(p) - f_0(p) > 0$ implies $b_p - {\Phi'}_{f_1}(p) > b_p - {\Phi'}_{f_0}(p)/(1 - \alpha) > 0$ and $f_1(p) - f_0(p) < 0$ implies $b_p - {\Phi'}_{f_1}(p) < b_p - {\Phi'}_{f_0}(p)/(1+\alpha) < 0$. Combining these observations, we conclude that the potential Ψ does not decrease. □

Lemma 3. *The flow f always remains feasible, i.e., at all times $f(e) \leq c(e)$ and $\Phi(e) \leq c(e)$ for all edges e.*

Proof. We prove that $\Phi(e) \leq c(e)$ by induction on the number of rounds. Initially, since the flow satisfies the capacity constraints, we have $f(e) \leq c(e)$. Consider any round in which the flow $f(e)$ (or equivalently $\Phi'(e)$) increases, which is possible only when $f(p)$ increases for some path $p \ni e$. Since $f(p)$ is increased only when $\Phi'(p) < b_p(1 - \alpha)$, we conclude that $\Phi'(e) < b_p(1 - \alpha)$ in the beginning of this round. Since the cost $\Phi'(e)$ increases by at most a factor of $1 + \frac{\alpha}{4}$ in any single round, the cost $\Phi'(e)$, whenever it increases, increases to at most $b_p \leq 1$. Since $\Phi(e) = \frac{c(e)}{\mu}\Phi'(e) < c(e) \cdot \Phi'(e)$, we conclude that $\Phi(e) \leq c(e)$ after the round. Thus the induction is complete and $\Phi(e) \leq c(e)$ always holds. From the definition of $\Phi(e)$, this implies that $f(e) \leq c(e)$ also holds always. □

Lemma 4. *If the network is unsaturated, then in $O(\tau_0)$ rounds it becomes saturated where*

$$\tau_0 = \max_{e \in E} \log_{1+\beta} \frac{c(e)}{\ddot{f}(e)} = \tilde{O}\left(\frac{\log(k\mu)}{\beta}\right). \tag{2}$$

Furthermore, once the network becomes saturated, it always remains saturated.

Proof. While the network is unsaturated, the cost of any path p satisfies $\Phi'(p) \leq \Phi' < \frac{1}{B} \cdot (1 - \alpha) \leq b_p \cdot (1 - \alpha)$. Thus all the flows increase by a factor of $(1 + \beta)$. Since after one round, the flow on any edge e is at least $\ddot{f}(e)$ and it never exceeds $c(e)$, the network has to become saturated in $O(\tau_0)$ rounds.

Now assume that the network becomes unsaturated again. Consider the round during which the network changes from being saturated to being unsaturated. In this round, the flow on some path p must decrease. Just before decrease, it must be true that $\Phi' \geq \Phi'(p) > b_p \cdot (1 + \alpha)$. However since single round reduces Φ' by at most $\frac{\alpha}{4}$ factor, we have $\Phi' > b_p \cdot (1 + \alpha)/(1 + \frac{\alpha}{4}) > \frac{1}{B}$ after the round. This is a contradiction. Thus we conclude that the network never becomes unsaturated again. □

The proofs of following two lemmas are omitted due to lack of space.

Lemma 5. *At all times, $(1 - 2\epsilon) \cdot \left(1 - \frac{1}{mCB}\right) \cdot |\ell_f| \cdot \Phi \cdot \mu \leq \Gamma \leq |\ell_f| \cdot \Phi \cdot \mu$.*

Lemma 6. *If the network is saturated, then $|\ell_f| \geq 1 - 2\epsilon$.*

Theorem 2. *Suppose that the network is either* cheap *or* expensive. *Then, in these cases Ψ increases by at least $\Delta\Psi = \Omega(\alpha\beta) \cdot (U + \Gamma)$ in a single round.*

Proof. From equation (1), we recall that the increase in Ψ in a single round is

$$\Delta\Psi = \Psi_1 - \Psi_0 \geq \sum_{p \in \mathcal{P}} (f_1(p) - f_0(p)) \cdot (b_p - \Phi'_{f_1}(p))$$

where the subscripts 0 and 1 indicate the values in the beginning and at the end of a round respectively. Denote $\mathcal{P}_x = \{p \in \mathcal{P} \mid \Phi'_{f_0}(p) \geq x \cdot b_p\}$ and $\mathcal{P}^y = \{p \in \mathcal{P} \mid \Phi'_{f_0}(p) \leq y \cdot b_p\}$ in the beginning of a round.

For every cheap path $p \in \mathcal{P}^{(1-\alpha)}$, we have $f_1(p) - f_0(p) \geq \beta f_0(p)$. For every expensive path $p \in \mathcal{P}_{(1+\alpha)}$, we have $f_1(p) - f_0(p) \leq -\beta f_0(p)$. Therefore

$$\begin{aligned}
\Delta\Psi \geq & \sum_{p \in \mathcal{P}^{(1-\alpha)} \cup \mathcal{P}_{(1+\alpha)}} (f_1(p) - f_0(p)) \cdot (b_p - \Phi'_{f_1}(p)) \\
\geq & \sum_{p \in \mathcal{P}^{(1-\alpha)}} \beta f_0(p) \cdot \left(b_p - \Phi'_{f_0}(p) \cdot \left(1 + \frac{\alpha}{4}\right)\right) \\
& \quad + \sum_{p \in \mathcal{P}_{(1+\alpha)}} -\beta f_0(p) \cdot \left(b_p - \Phi'_{f_0}(p) / \left(1 + \frac{\alpha}{4}\right)\right) & (3) \\
\geq & \frac{1}{1-\alpha} \sum_{p \in \mathcal{P}^{(1-\alpha)}} \beta f_0(p) \cdot ((1-\alpha) b_p - \Phi'_{f_0}(p)) \\
& \quad + \frac{1}{1+\alpha} \sum_{p \in \mathcal{P}_{(1+\alpha)}} -\beta f_0(p) \cdot ((1+\alpha) b_p - \Phi'_{f_0}(p)). & (4)
\end{aligned}$$

The inequality (3) follows from the fact that the cost of any edge (or path) changes in one round by at most $1 + \frac{\alpha}{4}$ factor. Note that both the terms in (4) are non-negative.

Cheap network: $\Gamma < (1-2\alpha) \cdot U$. From the definition of $\mathcal{P}^{(1-\alpha)}$, the 1st term in (4) is at least $\frac{1}{1-\alpha} \sum_{p \in \mathcal{P}} \beta f_0(p) \cdot ((1-\alpha) b_p - \Phi'_{f_0}(p)) = \frac{\beta}{1-\alpha}((1-\alpha)U - \Gamma) \geq \frac{\alpha\beta}{1-\alpha} \cdot U \geq \Omega(\alpha\beta) \cdot \Gamma$.

Expensive network: $\Gamma > (1+2\alpha) \cdot U$. From the definition of $\mathcal{P}_{(1+\alpha)}$, the 2nd term in (4) is at least $\frac{1}{1+\alpha} \sum_{p \in \mathcal{P}} -\beta f_0(p) \cdot ((1+\alpha) b_p - \Phi'_{f_0}(p)) = \frac{-\beta}{1+\alpha}((1+\alpha)U - \Gamma) \geq \Omega(\alpha\beta) \cdot \Gamma \geq \Omega(\alpha\beta) \cdot U$.

Putting things together, the proof is complete. □

Definition 3 (Reasonable interval). *We call an interval $\mathcal{T} = [t_0, t_1]$ of rounds* reasonable *if the network is reasonable at each round $t \in \mathcal{T}$. Otherwise, we call the interval* unreasonable*.*

4.2 Mileage definitions

We define the *offline mileage* $\nabla^{\Lambda}(e,t)$, the *derivative mileage* $\nabla^{\Phi'}(e,t)$, and benefit mileage $\nabla^{U}(t)$, as the absolute value of the change, that takes place in round t, in the offline potential, derivative cost function, and flow on edge e respectively. The second parameter t indicates that these values are in round t.

$$\nabla^{\Lambda}(e,t) = |\Lambda(e,t) - \Lambda(e,t-1)|, \qquad \nabla^{\Phi'}(e,t) = |\Phi'(e,t) - \Phi'(e,t-1)|.$$

For an interval $\mathcal{T} = [t_0, t_1]$ of rounds, we define $\nabla^{\Lambda}(e, \mathcal{T}) = \sum_{t\in\mathcal{T}} \nabla^{\Lambda}(e, t)$ and $\nabla^{\Phi'}(P, \mathcal{T}) = \sum_{e\in P}\sum_{t\in\mathcal{T}} \nabla^{\Phi'}(e, t)$. We also define total mileage for the whole network: $\nabla^{\Lambda}(t) = \sum_{e\in E} \nabla^{\Lambda}(e, t)$ and $\nabla^{U}(t) = |U(t) - U(t-1)|$. Also $\nabla^{\Lambda}(\mathcal{T}) = \sum_{t\in\mathcal{T}} \nabla^{\Lambda}(t)$ and $\nabla^{U}(\mathcal{T}) = \sum_{t\in\mathcal{T}} \nabla^{U}(t)$.

Theorem 3 lower bounds the increase $\Delta\Psi$ in the potential Ψ in an interval of rounds in terms of mileages of offline potential and flow respectively. This key theorem is proved in Section 5.

Theorem 3 (Potential Increase). *In any interval $\mathcal{T} = [t_0, t_1]$ the increase in potential is at least $\Delta\Psi > \Omega(\nabla^{\Lambda}(\mathcal{T}) \cdot \frac{\alpha}{\mu})$ and $\Delta\Psi > \Omega(\nabla^{U}(\mathcal{T}) \cdot \alpha)$.*

Definition 4 (Stationary interval). *We call an interval of time $\mathcal{T} = [t_0, t_1]$* stationary *if the mileage for offline potential and flow are small as compared to the flow volume at time t_0. Specifically, $\nabla^{\Lambda}(\mathcal{T}) \leq \frac{\alpha^2}{64} \cdot U(t_0)$, and $\nabla^{U}(\mathcal{T}) \leq \frac{\beta}{2} \cdot U(t_0)$. Otherwise, we call the interval* unstationary.

The corollary below follows directly from Theorem 3 and Definition 4.

Corollary 1. *Once the network is saturated, each* unstationary *interval $\mathcal{T} = [t_0, t_1]$, leads to a large increase in the potential, i.e., $\Delta\Psi > \Omega(\min\{\frac{\alpha^3}{\mu}, \alpha \cdot \beta\}) \cdot U(t_0)$.*

4.3 Main Theorems

Let U^* be the optimal flow value. The following theorem states that any sufficiently long stationary interval leads to near optimality. This key theorem is proved in Section 6.

Theorem 4 (Optimality). *Assume that the network is saturated. Consider a reasonable and stationary interval $\mathcal{T} = [t_0, t_1]$ of length at least τ_0 as in (2). At some round $t \in \mathcal{T}$, we get near optimality: $U(t) \cdot (1 + O(\beta + \epsilon)) \geq U^*$.*

Corollary 2 (Main). *After $O(\tau_\ddagger = \tau_0 \cdot \tau_\dagger)$ time, where τ_0 is as in (2) and*

$$\tau_\dagger = O(\epsilon \cdot \mu^4 \cdot \ln(mC)) = O(\epsilon \cdot \mu^5), \tag{5}$$

or, after $\tau_\ddagger = O(\epsilon \cdot \mu^7 \cdot \ln(k\mu)) = \tilde{O}\left(\frac{\log k \cdot \log^7(mCB)}{\epsilon^6}\right)$ time, the flow becomes near optimal, namely $U \cdot (1 + O(\beta + \epsilon)) \geq U^*$. Moreover, the total number of rounds in which the solution is not *$1 + O(\beta + \epsilon)$ approximate is also $\tilde{O}\left(\frac{\log k \cdot \log^{O(1)}(mCB)}{\epsilon^{O(1)}}\right)$.*

The proof of the above corollary follows from Corollary 1, Theorem 4, and the fact that the potential is bounded above by $m \cdot C$. The proof is omitted due to lack of space.

5 Potential increase proof of Theorem 3

Theorem 5 (Offline mileage). *If the offline mileage in an interval $\mathcal{T}$ is $\nabla^{\Lambda}(\mathcal{T})$, then Ψ increases in $\mathcal{T}$ by at least $\Omega(\nabla^{\Lambda}(\mathcal{T}) \cdot \frac{\alpha}{\mu})$.*

Proof. To simplify the intuition behind the proof, assume that the augmentation or withdrawal of flows on different paths happens *sequentially* (see the remark at the end of the proof).

Consider δ units of flow sent on path p. The offline mileage ∇^{Λ} due to this can be upper bounded by the increase $\Delta\Psi$ in the potential due to this as follows.

$$\frac{1}{\delta} \cdot \nabla^{\Lambda} = \Lambda'(p) = \sum_{e \in p} \Lambda'(f(e)) = \sum_{e \in p} \Phi''_e(f(e)) \cdot g(e) = \sum_{e \in p} \Phi'_e(f(e)) \cdot \frac{\mu}{c(e)} \cdot g(e)$$
$$\leq \sum_{e \in p} \Phi'_e(f(e)) \cdot \mu = \Phi'(p) \cdot \mu \leq O\left(\frac{\mu}{\alpha} \cdot \Psi'(p)\right) = O\left(\frac{1}{\delta} \cdot \frac{\mu}{\alpha} \cdot \Delta\Psi\right).$$

The first inequality holds since $g(e) \leq c(e)$. The second inequality above follows from the fact that $\Psi'(p) = b_p - \Phi'(p)$ and that the flow on p is increased only if $\Phi'(p) < (1-\alpha) \cdot b_p$, which in turn implies that $\Psi'(p) \geq \frac{1}{1-\alpha}\Phi'(p) - \Phi'(p) = \Omega(\alpha) \cdot \Phi'(p)$. A similar argument holds when δ units of flow is reduced on path p when $\Phi'(p) > (1+\alpha) \cdot b_p$. The overall offline mileage in a round is at most the sum of such mileage contributions over all the paths. Since each unit of offline mileage contributes to an increase of $\Omega(\frac{\alpha}{\mu})$ units of Ψ, the proof is complete.

Remark: The proof also works without the assumption of sequential execution, since the cost $\Phi'(f(e))$ of any edge changes by a factor of at most $1 + \frac{\alpha}{4}$ in a single round and $\Psi'(p) \geq \Omega(\alpha) \cdot \Phi'(p)$ still holds. □

Theorem 6 (Benefit mileage). *If the benefit mileage in an interval $\mathcal{T}$ is $\nabla^U(\mathcal{T})$, then Ψ increases in $\mathcal{T}$ by at least $\Omega(\nabla^U(\mathcal{T}) \cdot \alpha)$.*

Proof. Note that $\Psi'(p) = b_p - \Phi'(p)$ denotes the increase in Ψ per unit flow increase in $f(p)$. Observe that flow $f(p)$ is increased only along paths with $\Phi'(p) < (1-\alpha) \cdot b_p$ and is decreased only along paths p with $\Phi'(p) > (1+\alpha) \cdot b_p$. Thus, the net contribution of "positive" terms (namely, b_p for $\Delta f(p) > 0$ and $-\Phi'(p)$ for $\Delta f(p) < 0$) is at least $1 + \Omega(\alpha)$ factor higher that the net contribution of "negative" terms (namely, $-\Phi'(p)$ for $\Delta f(p) > 0$ and b_p for $\Delta f(p) < 0$). The net effect is at least α fraction of either change in the positive or negative terms. Thus any unit change in $f(p)$ contributes to $\alpha \cdot b_p$ increase in Ψ. Since the overall benefit mileage is at most the contributions of mileages of all the paths, the proof is complete. □

6 Optimality proof of Theorem 4

Suppose the network is saturated and the interval $\mathcal{T} = [t_0, t_1]$ of length at least τ_0 is both reasonable and stationary.

Lemma 7. *For all $t \in \mathcal{T}$, we have $\frac{\Gamma(t)-U(t)}{\Gamma(t)} \leq \frac{1}{2\mu}$.*

Proof. If $\Gamma(t) \leq U(t)$, then the above inequality trivially holds. On the other hand, since the network is *not* expensive at time t, we have $\Gamma(t) \leq U(t) \cdot (1+2\alpha)$. Therefore $\Gamma(t) - U(t) \leq 2\alpha \cdot U(t) \leq 2\alpha \cdot \Gamma(t)$. Since $\alpha = \frac{1}{4\mu}$, the lemma follows. □

Recall that g denotes the optimum flow that achieves the maximum throughput U^*. Let $h = g \cdot \frac{U(t_0)}{U^*}$ be the scaled optimum flow that achieves the throughput $U(t_0)$, the throughput of the solution f at time t_0. It is easy to see that if $|\ell_f| \leq (1 + O(\epsilon)) \cdot |\ell_h|$ at time t_0, where $|\ell_h|$ is the maximum edge load under flow h, then we have near optimality: $U(t_0) \geq (1 - O(\epsilon))U^*$.

Let $\hat{\Lambda}(t) = \sum_{e \in E} h(e) \cdot \Phi'(e) = \sum_{p \in \mathcal{P}} h(p) \cdot \Phi'(p)$ be the cost of flow h under the current cost metric. The next theorem shows that if $\Gamma(t)$ is a good approximation of $\hat{\Lambda}(t)$, then we have near optimality.

Theorem 7. *If at some time $t \in \mathcal{T}$, we have $\frac{\Gamma(t) - \hat{\Lambda}(t)}{\Gamma(t)} \leq \frac{7}{8\mu}$, then we have near optimality: $U(t) \cdot (1 + O(\beta + \epsilon)) \geq U^*$.*

Proof. Let $\Phi(h) = \sum_{e \in E} c(e) \cdot \exp[\mu(\frac{h(e)}{c(e)} - 1)]$ be the potential of h. From Lemma 1, we know that $\Phi(f(t)) - \Phi(h) \leq \Phi'(f(t)) \cdot (f(t) - h) = \Gamma(t) - \hat{\Lambda}(t)$. Since $\Gamma(t) \leq \Phi(t) \cdot \mu$, we conclude

$$\frac{\Phi(f(t)) - \Phi(h)}{\Phi(f(t))} \leq \frac{\Gamma(t) - \hat{\Lambda}(t)}{\Gamma(t)/\mu} \leq \frac{7}{8\mu} \cdot \mu = \frac{7}{8}.$$

This combined with the definition of Φ in turn implies that we have achieved additive ϵ approximation: $|\ell_f| \leq |\ell_h| + \epsilon$. Since the network is saturated, which implies $|\ell_f| \geq 1 - 2\epsilon$, we have multiplicative approximation $|\ell_f| \leq (1 + O(\epsilon)) \cdot |\ell_h|$.

Recall that the flow h has throughput $U(t_0)$, the throughput at the beginning of the interval. Since the benefit mileage is small: $\nabla^U(\mathcal{T}) \leq \frac{\beta}{2} \cdot U(t_0)$, we have $U(t) \geq (1 - \frac{\beta}{2}) \cdot U(t_0)$. Putting all things together we have near optimality: $U(t) \geq (1 - O(\epsilon + \beta)) \cdot U^*$. □

In light of Lemma 7 and Theorem 7, we can assume, without loss of generality, that for all $t \in \mathcal{T}$, $\frac{\hat{\Lambda}(t) - U(t)}{\Gamma(t)} = \frac{\hat{\Lambda}(t) - \Gamma(t)}{\Gamma(t)} + \frac{\Gamma(t) - U(t)}{\Gamma(t)} \leq \frac{-7}{8\mu} + \frac{1}{2\mu} = \frac{-3}{8\mu}$. Since the network is reasonable, $\Gamma(t) \geq (1 - 2\alpha) \cdot U(t)$. Therefore after simplifying, we get $\hat{\Lambda}(t) \leq U(t) \cdot \left(1 - \frac{3\alpha}{2} + 3\alpha^2\right)$. Since low benefit mileage implies $U(t) \leq U(t_0) \cdot (1 + \frac{\beta}{2})$, we conclude that the average cost of the flow path in h is small: $\frac{\hat{\Lambda}(t)}{U(t_0)} \leq \left(1 - \frac{3\alpha}{2} + 3\alpha^2\right) \left(1 + \frac{\beta}{2}\right)$.

Since $h = g \cdot \frac{U(t_0)}{U^*}$, we have that the mileage of h in the interval $\mathcal{T}$ (defined similarly and denoted by $\nabla^{\hat{\Lambda}}(\mathcal{T})$) is small: $\nabla^{\hat{\Lambda}}(\mathcal{T}) \leq \frac{\alpha^2}{64} \cdot U(t_0)$.

Lemma 8. *Consider a probability space $(\Omega, \pi : \Omega \rightarrow \Re^+)$ and two non-negative random variables χ and ψ with expectations $\bar{\chi}$ and $\bar{\psi}$ respectively. Then, for every $\kappa < 1$ there exists $\omega^* \in \Omega$ such that $\chi(\omega^*) \leq (1 + \kappa) \cdot \bar{\chi}$ and $\psi(\omega^*) \leq \frac{2}{\kappa} \cdot \bar{\psi}$.*

The above lemma applied to the set of paths $\mathcal{P}$ with a probability function π given by $\pi(p) = h(p) \cdot b_p / U(t_0)$, the two random variables given by $\chi(p) = \Phi'(p, t_0)/b_p$ and $\psi(p) = \nabla^{\Phi'}(p)/b_p$, and $\kappa = \alpha/4$, we get the following theorem.

Theorem 8 (Anchor theorem). *There exists a path $\mathbf{D} \in \mathcal{P}$ (referred to as "anchor" path) such that*

$$\frac{\Phi'(\mathbf{D}, t_0)}{b_{\mathbf{D}}} \leq (1 + \frac{\alpha}{4}) \cdot \frac{\hat{\Lambda}(t_0)}{U(t_0)} \quad \textit{and} \quad \frac{\nabla^{\Phi'}(\mathbf{D}, \mathcal{T})}{b_{\mathbf{D}}} \leq \frac{8}{\alpha} \cdot \frac{\nabla^{\Lambda}(\mathcal{T})}{U(t_0)}.$$

Using the above theorem, we get $\frac{\varPhi'(\mathbf{D},t_0)}{b_{\mathbf{D}}} \le \left(1+\frac{\alpha}{4}\right)\cdot\left(1-\frac{3\alpha}{2}+3\alpha^2\right)\left(1+\frac{\beta}{2}\right)$ and $\frac{\nabla^{\varPhi'}(\mathbf{D},\mathcal{T})}{b_{\mathbf{D}}} \le \frac{8}{\alpha}\cdot\frac{\alpha^2}{64} = \frac{\alpha}{8}$. Therefore we conclude that the cost $\varPhi'(\mathbf{D})$ at any time $t \in \mathcal{T}$ is at most

$$\begin{aligned}\varPhi'(\mathbf{D},t) &\le b_{\mathbf{D}}\cdot\left(\varPhi'(\mathbf{D},t_0)+\nabla^{\varPhi'}(\mathbf{D},\mathcal{T})\right)\\ &\le b_{\mathbf{D}}\cdot\left[\left(1-\frac{5\alpha}{4}+O(\alpha^2)\right)\cdot(1+O(\alpha^2))+\frac{\alpha}{8}\right] < b_{\mathbf{D}}\cdot(1-\alpha).\end{aligned}$$

This leads to a contradiction as follows. Since the cost of the anchor path $\mathbf{D}$ is *consistently* lower than $b_{\mathbf{D}}\cdot(1-\alpha)$, its flow $f(\mathbf{D})$ is increased by a factor $1+\beta$ in every single round. Thus after τ_0 rounds, the flow on some edge in $\mathbf{D}$ must exceed its capacity. This is a contradiction and thus the proof of Theorem 4 is complete. □

References

1. B. Awerbuch, Y. Azar, and R. Khandekar. Fast load balancing via bounded best response. *SODA*, 2008.
2. B. Awerbuch and R. Khandekar. Distributed network monitoring and multicommodity flows:primal-dual approach. *PODC*, 2007.
3. B. Awerbuch and R. Khandekar. Greedy distributed optimization of multi-commodity flows. *PODC*, 2007.
4. B. Awerbuch, R. Khandekar, and S. Rao. Distributed algorithms for multicommodity flow problems via approximate steepest descent framework. *SODA*, 2007.
5. B. Awerbuch, B. Patt-Shamir, and G. Varghese. Self-stabilization by local checking and correction. *FOCS*, 1991.
6. B. Awerbuch and G. Varghese. Distributed program checking: a paradigm for building self-stabilizing distributed protocols. *FOCS*, 1991.
7. Y. Bartal, J. W. Byers, and D. Raz. Global optimization using local information with applications to flow control. *FOCS*, 1997.
8. E. Dijkstra. Self stabilizing systems in spite of distributed control. *CACM*, 17:643–644, Nov 1974.
9. S. Dolev, A. Israeli, and S. Moran. Self-stabilization of dynamic systems assuming only read/write atomicity. *PODC*, 1990.
10. L. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM Journal on Discrete Mathematics*, 13:505–520, 2000.
11. N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *FOCS*, 1998.
12. N. Garg and N. E. Young. On-line, end-to-end congestion control. *FOCS*, 2002.
13. M. G. Gouda and N. J. Multari. Stabilizing communication protocols. Technical Report TR-90-20, Dept. of Computer Science, University of Texas at Austin, June 1990.
14. C. Koufogiannakis and N. E. Young. Beating simplex for fractional packing and covering linear programs. *FOCS*, 2007.
15. F. Kuhn. *The price of locality: exploring the complexity of distributed coordination primitives.* PhD Thesis, ETH Zurich, Diss. ETH No. 16213, December 2005.
16. M. Luby and N. Nisan. A parallel approximation algorithm for positive linear programming. *STOC*, 1993.
17. S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math of Oper. Research*, 20(2), pages 257–301, 1994.
18. N. E. Young. Sequential and parallel algorithms for mixed packing and covering. *FOCS*, 2001.