Local routing algorithms for node-capacitated multicommodity broadcast

Andrew Twigg and Laurent Massoulié
Thomson Research, Paris
firstname.lastname@thomson.net

Thomson Technical Report
Number: CR-PRL-2007-08-0001
Date: August 22, 2007

Abstract: Given a directed network with upload and download capacities at nodes, we consider the multicommodity relay-free multicast problem: there are k multicast sessions, each with a source $s_i \in V$, receiver set $R_i \subseteq V$ and demand $\lambda_i > 0$, with the constraint that nodes only forward packets of commodities for which they are also a receiver. When each R_i is a clique, we prove that whenever demands $\{\lambda_i + \varepsilon\}$ are feasible, a simple local-control routing algorithm is stable under demands $\{\lambda_i\}$. We also give a randomized procedure for resampling neighbours that allows the use of bounded-degree neighbourhoods.

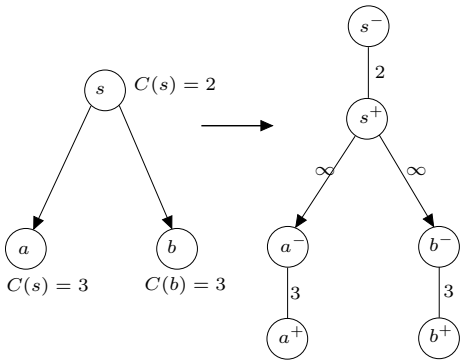

Figure 1: An example of when the simple node-capacity transformation used in the unicast case fails in the broadcast case

1 Introduction

Given a directed network $G = (V, E)$ with n nodes, a multicast session (s_i, R_i, λ_i) contains a source $s_i \in V$, a receiver set $R_i \subseteq V$ and a nonnegative demand λ_i. We assume that $s_i \in R_i$. We consider the node-capacitated problem, where each node u has upload capacity c_u^+ and download capacity c_u^-. We are interested in local-control algorithms to route the commodities from sources to receiver sets that are stable whenever the demands are feasible.

When there is a single receiver set containing a single destination node, we have the node-capacitated unicast problem. In this case, Menger's theorem [5] says that there are k vertex-disjoint paths between a source s and sink t if and only if G is k-vertex-connected, and the maxflow algorithm of Ford and Fulkerson [2] can be used (replacing each node of capacity c by two nodes linked by an edge of capacity c).

When the receiver sets contain several nodes, the above technique cannot be used. Consider the network in Figure 1 – the node-capacitated network on the left can clearly only support a broadcast rate of one, but the min-mincut of the network on the right is two (and by Edmonds' theorem [1] has broadcast rate two). The problem arises because the transformation 'creates' capacity when there are multiple receivers sharing the same flow paths.

1.1 Contributions

In recent work [4], we showed that a simple local-control algorithm achieves the optimal broadcast rate for the complete graph, when only upload capacities are considered (download capacities were assumed to be infinite) and there is a single commodity (s, V, λ) to be broadcast to all nodes.

In this paper, we consider the multicommodity version of the problem with a natural restriction that there are no relay nodes, i.e. node u will only forward packets of commodities i for which $u \in R_i$. This is a natural restriction to consider when one cannot assume that network users will forward packets for commodities for which they have no interest in receiving. Indeed, this is the current model employed by applications such as BitTorrent. We also consider the case where download capacities are limited, and show that the same local-control algorithm achieves the optimal broadcast rate whenever a given set of demands is feasible.

The paper is structured as follows. In Section 2 we describe the algorithm and derive the fluid limits that we shall make use of in the remainder. In Section 4 we consider the case where nodes have restrictions on both their download and upload capacities, and where there is a single commodity. In Section 5 we consider the multicommodity case with no download constraints. In Section 6 we describe a resampling procedure that allows us the use of low degree neighbourhoods while maintaining optimal throughput. The techniques presented in these sections can be combined to get a stronger result, but for clarity we present them separately.

2 Preliminaries

We consider the following model. Let I be the set of all commodities. Each node u has a neighbourhood set $N(u)$ and $|I|+1$ buffers; for each commodity $i \in I$, a collection $P^i(u)$ that stores the packets of commodity i it has received, and a separate input buffer. When a packet of commodity i is transmitted to u, it enters u's input buffer. Packets at the input buffer are served uar with service time exponentially distributed having mean $\mu_u = 1/c_u^-$. When a packet is served, it leaves the input buffer and enters the collection $P^i(u)$. Packets do not leave the collection $P^i(u)$ (we assume that nodes want to store the data they are interested in receiving). This is illustrated in Figure 2.

We consider the following algorithm. Let $R(u) = \{i : u \in R_i\}$ be the commodities for which u is a receiver, and $R(uv) = R(u) \cap R(v)$. We shall

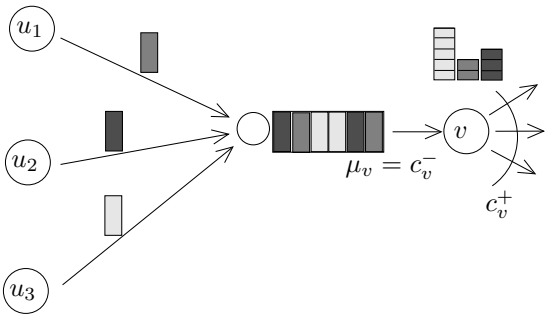

Figure 2: Illustrating the model used. Each node v has a single input buffer, with service time exponentially distributed having mean μ_v and one output buffer per commodity for which it is a receiver.

denote by G_i the subgraph of G induced by R_i. For nodes u, v, define

$$P_{+u-v} = \bigcup_{i \in R(uv)} P^i(u) \setminus P^i(v)$$

as the collection of packets that u can forward to v, and that v is interested in receiving. Each non-source node u chooses a node $v \in N(u)$ s.t. $|P_{+u-v}|$ is maximal, and sends at rate c_u^+ a packet chosen uar from P_{+u-v}. A source node s_i chooses a neighbouring node v in the same way, but if there is a packet in P_{+s_i-v} that s_i has not yet sent to any node (a *fresh packet*), this packet is sent (if there are several fresh packets, one is chosen uar), otherwise it sends a packet uar from P_{+s_i-v} to v. For convenience, let $P^i_{\text{fresh}}(s_i)$ be the set of fresh packets at s_i.

RANDOM-USEFUL-MOST-DEPRIVED(RMD)(u)

1 At rate $C(u)$
2 **do** $v \leftarrow$ random neighbor maximizing $|P_{+u-v}|$
3 **if** $u = s_i$ for some $i \in I$ and $P^i_{\text{fresh}}(s_i) \neq \emptyset$
4 **then** Send a random packet in $P^i_{\text{fresh}}(s_i)$ to v
5 **else** Send a random packet in P_{+u-v} to v

3 Markov process and fluid limits

In order to evaluate the performance of our algorithm, we shall first examine the lifetime of a typical packet. Once injected at source s_i, a packet p can

be in a number of different states: *(a)* It can be replicated at all nodes in R_i, hence successfully broadcast. *(b)* It can be idle, that is not actively transferred, and be replicated at nodes u in some set $S \subset R_i$. The subset S cannot be arbitrary; it must contain a spanning tree of $G_i \cap S$ rooted at s_i. *(c)* It can be replicated at some nodes $u \in S$ for some subset S, and also actively transferred along some edges $e \in F$, for some subset $F \subseteq (S, \bar{S})$.

We shall describe the state of the system as follows: *(a)* For all $S \subset V$, X_S^i denotes the number of idle packets of commodity $i \in I$ that are replicated exactly at the nodes $u \in S$. *(b)* For each commodity $i \in I$, $A^i = \{G_1 = (W_1, F_1), \ldots, G_m = (W_m, F_m)\}$ is an unordered list of subgraphs which describes the active packets of commodity i. W_j denotes the set of nodes at which the jth active packet is currently replicated; F_j is the set of edges along which the jth active packet is currently transferred.

Since each node forwards a packet to only one of its neighbours at a given time. Thus for each node u, there is at most one edge (u, w) appearing in the sets F_i, $i = 1, \ldots, m$. We shall assume that packet transmissions are not preempted, even if a neighbour of some node u becomes more deprived than the neighbour v to which node u is currently transmitting.

We shall assume that at any given time, at most one packet is transferred from a given node, hence, the total number of active packets is (at most) $|V| = n$. We shall further assume that the same active packet cannot be received from multiple incoming edges. We also enforce an *activity condition* which states that if there is no transfer from u then necessarily there is no packet that could be transferred from u to any of its neighbours.

For commodity i, we assume that new packets are injected into each source s_i at instants of independent Poisson processs having rate λ_i. We also assume that the packet transfer time along an edge (uv) is exponentially distributed with mean $1/c_{uv}$. The process described by the states $\left(\left(X_S^i\right)_{S\subset V, i\in I}, \left(A^i\right)_{i\in I}\right)$ and the state transitions will be crucial for analyzing the performance of the algorithm.

3.1 Fluid limits

We refer the reader to the full paper for the detailed description of the Markov process and the derivation of fluid limits from rescaling the Markov process in space and time as in [4]. Due to space constraints, we simply state the fluid limits without proof.

We begin by setting our notation. For a commodity i and set $S \subset V$, let $y_S^i(t)$ be a quantity representing the amount of commodity i replicated exactly at nodes in S at time t. For a subset $S \subset V$, we write $y_{\subseteq S}^i =$

$\sum_{S' \subseteq S} y^i_{S'}$, and for a set of commodities T, $y^T_S = \sum_{i \in T} y^i_S$. We define the *potential* of commodity i between u and v as $y^i_{+u-v} = \sum_{S:(uv) \in (S,\bar{S})} y^i_S$, and $y^T_{+u-v} = \sum_{i \in T} y^i_{+u-v}$ for a set of commodities T. For simplicity, we shall sometimes write $X + i$ to mean $X \cup \{i\}$ for set X and element i.

Definition 1 (Fluid trajectories) *The real-valued nonnegative functions $y^i_S(t), S \subset V, i \in I$, are called fluid trajectories of the Markov process if they satisfy the following conditions. For all $S \subset V, u \in S, v \notin S, i \in I$, there exist nonnegative functions $\phi^i_{S,(uv)}(t)$ that are non-decreasing such that:*

$$y^i_{\{s_i\}}(t) \quad = \quad y^i_{\{s_i\}}(0) + \lambda_i t - \textstyle\sum_{v \in V \setminus \{s_i\}} \phi^i_{\{s_i\},(s_i v)}(t) \tag{1}$$

and for $j \in I, j \neq i$ we have $y^j_{\{s_i\}}(t) = 0$. In addition, for $S \neq \{s_i\}$, we have

$$y^i_S(t) = y^i_S(0) + \sum_{u \in S, v \in S-u} \phi^i_{S-v,(uv)}(t) - \sum_{(uv) \in (S,\bar{S})} \phi^i_{S,(uv)}(t). \tag{2}$$

For all nodes u, the functions $\{\phi^i_{S,(uv)}(t)\}_{v:(uv) \in E, i \in R(uv), S \subset R_i, v \notin S}$ are differentiable at almost every t, otherwise we fix $\phi^i_{S,(uv)}(t) = 0$ for all other ϕ.

If $\sum_{v:(u,v) \in E} \sum_{i \in R(uv)} y^i_{+u-v}(t) > 0$, the derivatives satisfy

$$\frac{d}{dt} \phi^{R(uv)}_{S,(uv)}(t) = 0 \text{ if } y^{R(uv)}_{+u-v}(t) < \max_{v':(u,v') \in E} \left(y^{R(uv')}_{+u-v'}(t) \right), \tag{3}$$

$$\sum_{v:(uv) \in E} \sum_{S:u \in S, v \notin S} \frac{d}{dt} \phi^{R(uv)}_{S,(uv)}(t) = c_u \tag{4}$$

where we use the shorthand $\phi^T_{S,(uv)} = \sum_{i \in T} \phi^i_{S,(uv)}$.

If $u \neq s_i$ for some $i \in I$, that is for a non-source node, we have, for all v such that $(uv) \in E$ and assuming that $\sum_{S:u \in S, v \notin S} \sum_{i \in R(uv)} \frac{d}{dt} \phi^i_{S,(uv)}(t) > 0$ holds,

$$\begin{gathered} \forall i \in R(uv), S \subset R_i, u \in S, v \notin S : \\ \tfrac{d}{dt} \phi^i_{S,(uv)}(t) = \tfrac{y^i_S(t)}{y^{R(uv)}_{+u-v}(t)} \textstyle\sum_{S':u \in S', v \notin S'} \tfrac{d}{dt} \phi^{R(uv)}_{S',(uv)}(t) \end{gathered} \tag{5}$$

For a source node s_i with fresh packets to send, one has the following:

$$y^i_{\{s_i\}}(t) > 0 \Rightarrow \sum_{v \in R_i, v \neq s_i} \frac{d}{dt} \phi^i_{\{s_i\},(s_i v)}(t) = c_{s_i}. \tag{6}$$

In the case where the source s_i has no fresh packets, i.e. $y_{\{s\}} = 0$, we have for all v such that $(sv) \in E$, assuming that $\sum_{S \subset R_i : S \neq \{s_i\}, v \notin S} \frac{d}{dt}\phi^i_{S,(s_i v)}(t) > 0$ holds,

$$\begin{array}{c} \forall S \subset R_i, S \neq \{s_i\}, v \notin S : \\ \frac{d}{dt}\phi^i_{S,(s_i v)}(t) = \frac{y^i_S(t)}{\sum_{S' \subset R_i, S' \neq \{s_i\}, v \notin S'} y^i_{S'}(t)} \sum_{S' \subset R_i, S' \neq \{s_i\}, v \notin S'} \frac{d}{dt}\phi^i_{S',(s_i v)}(t) \end{array} \tag{7}$$

For non-source nodes, the intution is as follows: $\frac{d}{dt}\phi^i_{S,(uv)}(t)$ represents the rate at which packets of commodity i, previously replicated at nodes in S are replicated along edge (uv). This rate is the probability of such a packet being pushed along edge (uv), given by $y^i_S(t)/y^{R(uv)}_{+u-v}(t)$, multiplied by the rate at which packets are currently traversing edge (uv), which is $\sum_{S' : u \in S', v \notin S'} \frac{d}{dt}\phi^{R(uv)}_{S',(uv)}(t)$.

4 Upload and download capacities

In this section we consider the case where there is a single commodity (s, V, λ), G is the complete graph and each node u finite download and upload capacities. The restriction on a single commodity can be relaxed by using the results of Section 5 but significantly complicates the analysis.

First, given the complete graph $G = (V, E)$ with n nodes V having upload capacities c^+_u and download capacities c^-_u as described earlier, construct a graph G' by replacing each node u by two nodes u^-, u^+ as follows. Let $G' = (V^- \cup V^+, E')$ where $V^- = \{u^- : u \in V\}$ and similarly for V^+, and $E' = \{(u^- u^+) : u \in V\} \cup \{(u^+ u^-) : u, v \in V\}$. This graph G' has only upload capacity constraints given by $c_{u^-} = c^-_u$ and $c_{u^+} = c^+_u$. It can be seen that the algorithm is stable on G' (with only upload constraints) iff it is stable on G (with download and upload constraints), since the node c_{u^-} acts as the download buffer for node u. We assume that $c_s = c^+_s$ and $c^-_s = \infty$. The main difficulty is in showing stability of the algorithm on this non-complete graph G'.

Theorem 1 *Assume that demand λ satisfies*

$$\begin{array}{rcl} \lambda + \varepsilon & \leq & c^+_s \\ \lambda + \varepsilon & \leq & \min_{u \in V} c^+_u \\ \lambda + \varepsilon & \leq & \min_{u \in V} c^-_u \end{array} \tag{8}$$

for some $\varepsilon > 0$. Then the algorithm is stable under demand λ.

By stable, we mean that the number X_t of packets undelivered to some receiver at time t converges in distribution as $t \to \infty$, hence is bounded in probability, i.e.

$$\exists \psi(a), \lim_{a\to\infty} \psi(a) = 0 \text{ s.t. } \Pr(X_t \geq a) \leq \psi(a), \forall t.$$

Note that whenever demand λ is feasible, a weaker version of condition (8) holds with the second requirement replaced by $\lambda + \varepsilon \leq \frac{\sum_{u\in V} c_u^+}{n-1}$. We have been unable to prove stability of the distributed algorithm under this assumption, but we believe it is possible to do so and we leave it as an open problem.

To show stability of the fluid trajectories, we rely on the following lemma.

Lemma 1 *For any $y = (y_S)_{S\in\mathcal{S}} \in \mathbb{R}_+^{\mathcal{S}}$, define the* workload *function $w(y)$ as:*

$$w(y) = \sum_{S\in\mathcal{S}} y_S \left(n - |S|\right). \tag{9}$$

Under the assumption (8), when the graph G is complete, any fluid trajectory y as per Definition 1 is such that, for some $\epsilon > 0$,

$$w(y(t)) \leq \max(0, w(y(0)) - \epsilon t). \tag{10}$$

Proof. First, note that the n in the lemma is the size of the graph G, so the size of the transformed graph is $n' = 2n$. The proof strategy is as follows. Transform the graph G having upload and download constraints into the equivalent G' having only upload constraints. Then there are two main cases to consider.

Case 1 Assume that all $u^- \in V^-$ are such that $y_{+(u^-)-(u^+)}(t) > 0$. Then consider two cases:

Case 1a If there is no node $u^+ \in V^+$ such that $\sum_{v^- \in V^-} y_{+(u^+)-(v^-)}(t) = 0$, then all nodes are doing useful work and the original argument of [4] gives

$$\begin{aligned}
\frac{d}{dt}w(y(t)) &\leq \lambda(n'-1) - \sum_{u \in V^- \cup V^+} c_u \\
&= \lambda(n'-1) - \sum_{u \in V} c_u^- - \sum_{u \in V} c_u^+ \\
&\leq \lambda(2n-1) - n\lambda - \sum_{u \in V} c_u^+ \\
&= \lambda(n-1) - \sum_{u \in V} c_u^+ \\
&< 0
\end{aligned}$$

under the assumptions (8), recalling that $n' = 2n$.

Case 1b Otherwise, there exists a set $S^* \subseteq V^+$ such that

$$\forall u^+ \in S^*, \sum_{v^- \in V^-} y_{+(u^+)-(v^-)}(t) = 0.$$

Since the only node that can send packets to a deprived $u^+ \in S^*$ is its corresponding neighbour u^-, we have to show that u^- does enough work to decrease the work function $w(y)$.

We can assume wlog that $y_{\{s\}}(t) = 0$ and hence $\frac{d}{dt}y_{\{s\}}(t) = 0$ by using the same continuity argument used in [4] (for otherwise, if $y_{\{s\}}(t) > 0$ then the source has fresh packets and the work function $w(y)$ would immediately be decreasing). This implies that $\sum_{v \in V'} \phi_{\{s\},(sv)}(t) = \lambda$ by examining the fluid trajectories. We also have, $\forall S \in \mathcal{S}$, if $S \cap S^* \neq \emptyset$ then $y_S(t) = 0$ and

hence $\frac{d}{dt}y_S(t) = 0$. Using this, write

$$\begin{aligned}
\frac{d}{dt}w(y(t)) &= \sum_{\substack{S\in\mathcal{S},S\neq\{s\}\\ S\cap S^*=\emptyset}} (2n-|S|)\frac{d}{dt}y_S(t) \\
&\leq (2n-1)\sum_{v\in V^-}\frac{d}{dt}\phi_{\{s\},(sv)}(t) - \sum_{\substack{S\in\mathcal{S},S\neq\{s\},\\ S\cap S^*=\emptyset}}\sum_{\substack{u\in V^+\cap S\\ v\in V^-,v\notin S}}\frac{d}{dt}\phi_{S,(uv)}(t) \\
&\quad - \sum_{\substack{S\in\mathcal{S},S\neq\{s\},\\ S\cap S^*=\emptyset}}\sum_{\substack{u\in V^-\cap S\\ v\in V^+,v\notin S,v\notin S^*}}\frac{d}{dt}\phi_{S,(uv)}(t) \\
&\quad - \sum_{\substack{S\in\mathcal{S},S\neq\{s\},\\ S\cap S^*=\emptyset}}(2n-|S|)\sum_{\substack{u\in V^-\cap S\\ v\in V^+,v\notin S,v\in S^*}}\frac{d}{dt}\phi_{S,(uv)}(t) \\
&\leq \lambda(2n-1) - \sum_{u\in V^+\setminus S^*}c_u^+ - \sum_{u:u^+\notin S^*}c_u^- - |S^*|\sum_{u^+\in S^*}c_u^- \\
&\leq \lambda(n-1+|S^*|-|S^*|^2) - \sum_{u:u^+\in V^+\setminus S^*}c_u^+ \\
&< 0
\end{aligned}$$

whenever $c_u^+ > \lambda$ for all u.

Case 2 The final case to consider is when there is a set $S^* \subseteq V^-$ of deprived nodes, where each $v^- \in S^*$ has $y_{+(v^-)-(v^+)}(t) = 0$. Since the source s connects directly to all the nodes in V^-, we can re-use exactly the same arguments as before, distinguishing between the cases $y_{\{s\}}(t) = 0$ and $y_{\{s\}}(t) > 0$, to show that the source alone does enough useful work to make $w(y)$ decrease. These give

$$\begin{aligned}
\frac{d}{dt}w(y(t)) &\leq -(c_s-\lambda) \\
&< 0
\end{aligned}$$

under the assumptions (8). This proves Lemma 1. □

Proof of Theorem 1. Theorem 1 can be proved by combining Lemma 1 with the following suitable ergodicity criterion, which appears as Theorem 8.13, p.224 in Robert [6].

Theorem 2 *Let $Z(t)$ be a Markov jump process on a countable state space $\mathcal{Z}$. Assume there exists a function $L : \mathcal{Z} \to \mathbb{R}_+$ and constants M, ϵ, $\tau > 0$ such that for all $z \in \mathcal{Z}$:*

$$L(z) > M \Rightarrow \frac{1}{L(z)} \mathbf{E_z L}\left(\mathbf{Z(L(z)}\tau)\right) \leq \mathbf{1} - \epsilon. \tag{11}$$

If in addition the set $\{z : L(z) \leq M\}$ is finite, and $\mathbf{E_z L(Z(1))} < +\infty$ for all $z \in \mathcal{Z}$, then the

Since the Markov process is ergodic, the number of packets X_t remaining undelivered to some receiver time t converges to a nondegenerate limiting distribution as $t \to \infty$, and hence is bounded in probability (though not necessarily in expectation). □

5 Multicommodity demands

In this section we consider the multicommodity case, and assume that download capacities $c_u^- = \infty$. The restriction on infinite download capacities can be relaxed by using the results of Section 4 but significantly complicates the analysis, so we omit it for clarity.

We shall prove optimality under the assumption that the subgraph $G[R_i]$ induced by each receiver set R_i is a clique. This is a relaxation of the assumption that the entire graph G is a clique. We also require that all nodes are either receivers of possibly several commodities, or a source of exactly one commodity. We believe it is possible to remove this last constraint, but do not try to do so here for clarity of the analysis. The fluid limits are those given in Definition 1.

The main result of this section is the following.

Theorem 3 *Assume that all nodes are either receivers or sources of exactly one commodity and that each receiver set induces a clique in G. Then whenever demands $\{\lambda_i + \varepsilon\}$ are feasible, for some $\varepsilon > 0$, the random-useful forwarding algorithm is stable under demands $\{\lambda_i\}$.*

As a corollary of the proof, we obtain the following tight characterisation of the set of feasible demands. For a set of commodities $J \subseteq I$, define $R_J = \bigcup_{i \in J} R_i$.

Corollary 1 *Demands $\{\lambda_i\}$ are feasible if and only if the following holds:*

$$\begin{array}{rcll} \lambda_i & \leq & c_{s_i} & \forall i \in I \\ \sum_{i \in J} \lambda_i & \leq & \frac{\sum_{u \in R_J} c_u}{|R_J| - |J|} & \forall J \subseteq I \end{array} \tag{12}$$

Whenever demands $\{\lambda_i\}$ are feasible, Equation (12) holds. Assume the demands are feasible but the condition does not hold. Then either there is a source with less capacity than its demand, or there is a set $J \subset I$ of commodities such that the subgraph of G induced by R_J does not have enough *total* upload capacity to support the transmission of commodities J, even assuming that packets from two commodities $i, j \in J$ are treated as if they are a single commodity. The denominator correctly counts the number of nodes that want to receive packets of commodity J, since each commodity has exactly one source, which is not present in any other receiver set. Clearly, this implies that the demands are not feasible.

To show stability of the fluid trajectories, we rely on the following lemma.

Lemma 2 *For any $y = (y_S)_{S\in\mathcal{S}} \in \mathbb{R}_+^{\mathcal{S}}$, define the* workload *function $w(y)$ as:*

$$w(y) = \sum_{i\in I} \sum_{S\in\mathcal{S}} y_S^i \left(|R_i| - |S|\right). \tag{13}$$

Assume that both the conditions (12) and the conditions of Theorem 3 are satisfied. Then any fluid trajectory y as per Definition 1 is such that, for some $\epsilon > 0$,

$$w(y(t)) \leq \max(0, w(y(0)) - \epsilon t). \tag{14}$$

Proof. Let $|I| = k$ and assume wlog that $|R_I| = n$ (otherwise, there are some nodes that will never participate in forwarding packets). Also assume wlog that each node is either a source of exactly one commodity or a receiver (of possibly several commodities), which can be handled by a similar transformation as in the multicommodity edge-capacitated case.

If $\forall u \in V, \sum_{v\neq u} \sum_{i\in R(uv)} y^i_{+u-v} > 0$ then every node is doing useful work, and it can be shown as in [3] that $w(y)$ decreases whenever

$$\sum_{i\in I} \lambda_i < \frac{\sum_{u\in V} c_u}{n-1}$$

holds, which is implied by the condition (12).

Otherwise, there exists a set $S^* \subseteq V$ such that

$$\forall u \in S^*, \sum_{v\neq u} \sum_{i\in R(uv)} y^i_{+u-v} = 0. \tag{15}$$

Define I^* as the set of commodities $I^* = \{i \in I : R_i \cap S^* \neq \emptyset\}$ having some deprived node in its receiver set. Define

$$\Theta(S^*) = \{v \in V : \exists u \in S^*, R(uv) \neq \emptyset\}$$

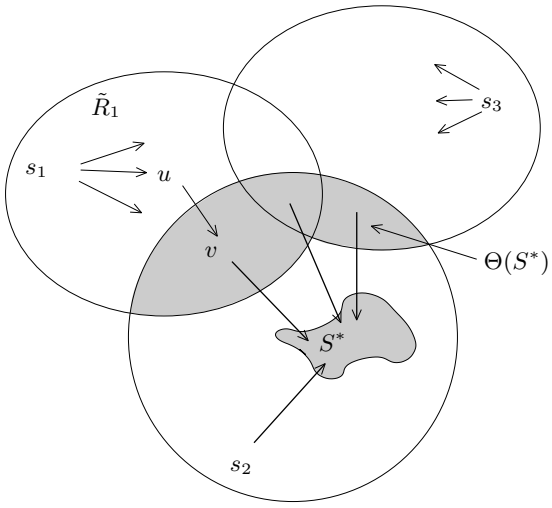

Figure 3: Illustrating the structure of the sets $\tilde{R}_i, \Theta(S^*)$. Every node $u \in R_i \setminus \tilde{R}_i$ devotes its entire capacity towards the nodes v in S^* with $R(uv) \neq \emptyset$.

as the 'neighbours' of S^*, so that $\forall u \in \Theta(S^*)$, u has a neighbour $v \in S^*$ that will be its 'most-deprived' neighbour, i.e. maximizing $|P_{+u-v}|$. Finally, for each $i \notin I^*$, define the subset $\widetilde{R}_i = R_i \setminus \Theta(S^*)$ (certainly, one must have $s_i \in \widetilde{R}_i, \forall i \notin I^*$). Figure 3 illustrates the structure of these sets.

Now, write

$$\frac{d}{dt}w(y(t)) = \underbrace{\sum_{i\in I^*}\sum_{S\subset R_i}\frac{d}{dt}y^i_S(|R_i|-|S|)}_{(A)} + \underbrace{\sum_{i\notin I^*}\sum_{S\subset R_i}\frac{d}{dt}y^i_S(|R_i|-|S|)}_{(B)}. \quad (16)$$

Now we tackle the first term (A). Note that $\forall S \subset V$ with $u \in S \cap S^*$, one has $y^i_S(t) = 0, \forall v \neq u, i \in R(uv)$, and hence $\frac{d}{dt}y^i_S(t) = 0$ by a similar argument as before.

Define $F = \{i \in I^* : y^i_{\{s_i\}}(t) > 0\}$ as the set of commodities in I^* whose sources have fresh packets to send. With this in hand, write

$$(A) = \underbrace{\sum_{i\in I^*\setminus F}\sum_{\substack{S\subset R_i, S\neq\{s_i\}\\ S\cap S^*=\emptyset}}(|R_i|-|S|)\frac{d}{dt}y^i_S}_{(A1)} + \underbrace{\sum_{i\in F}\sum_{S\subset R_i}(|R_i|-|S|)\frac{d}{dt}y^i_S}_{(A2)}. \quad (17)$$

For (A1), write

$$\begin{aligned}(A1) &= -\sum_{i\in I^*\setminus F}\sum_{\substack{S\subset R_i, S\neq\{s_i\}\\ S\cap S^*=\emptyset}}(|R_i|-|S|)\sum_{u\in S, v\in S^*}\frac{d}{dt}\phi^i_{S,(uv)}\\ &\leq -\sum_{i\in I^*\setminus F}\sum_{\substack{S\subset R_i, S\neq\{s_i\}\\ S\cap S^*=\emptyset}}\left(\sum_{v\in S^*}\frac{d}{dt}\phi^i_{S,(s_iv)}+\sum_{u\in S,u\neq s_i}\sum_{v\in S^*}\frac{d}{dt}\phi^i_{S,(uv)}\right)\\ &\leq -\sum_{i\in I^*\setminus F}(c_{s_i}-\lambda_i)-\sum_{i\in I^*\setminus F}\sum_{\substack{S\subset R_i, S\neq\{s_i\}\\ S\cap S^*=\emptyset}}\sum_{u\in S,u\neq s_i}\sum_{v\in S^*}\frac{d}{dt}\phi^i_{S,(uv)}\end{aligned}$$

where the final inequality uses the fact that $\forall i \in I^* \setminus F$, $\frac{d}{dt}y^i_{\{s_i\}} = 0 = \lambda_i - \sum_{v\in S^*}\frac{d}{dt}\phi^i_{\{s_i\},(s_iv)}$, hence

$$\sum_{\substack{S\subset R_i, S\neq\{s_i\}\\ S\cap S^*=\emptyset}}\sum_{v\in S^*}\frac{d}{dt}\phi^i_{S,(s_iv)} = c_{s_i}-\lambda_i.$$

For (A2), recall that every node is either a source of exactly one commodity or is a receiver. In this case for any i such that $y^i_{\{s_i\}}(t) > 0$, we have $\frac{d}{dt}y^i_{\{s_i\}} = \lambda_i - c_{s_i}$ by the fluid trajectories of sources with fresh packets to send. Hence we can write, in a similar way to the single-commodity case [3],

$$\begin{aligned}(A2) &= \sum_{i\in F}(|R_i|-1)(\lambda_i-c_{s_i})-\sum_{i\in F}\sum_{\substack{S\subset R_i, S\neq\{s_i\}\\ S\cap S^*=\emptyset}}(|R_i|-|S|)\sum_{u\in S, v\in S^*}\frac{d}{dt}\phi^i_{S,(uv)}\\ &< \sum_{i\in F}(\lambda_i-c_{s_i})-\sum_{i\in F}\sum_{\substack{S\subset R_i, S\neq\{s_i\}\\ S\cap S^*=\emptyset}}\sum_{u\in S, v\in S^*}\frac{d}{dt}\phi^i_{S,(uv)}\end{aligned}$$

where the final inequality follows under the condition (12).

Combining (A1) and (A2) gives

$$\begin{aligned}
(A) &\leq -\sum_{i\in I^*}(c_{s_i}-\lambda_i) - \sum_{i\in I^*}\sum_{\substack{S\subset R_i, S\neq\{s_i\}\\ S\cap S^*=\emptyset}}\sum_{u\in S, u\neq s_i}\sum_{v\in S^*}\frac{d}{dt}\phi^i_{S,(uv)}\\
&\leq -\sum_{i\in I^*}(c_{s_i}-\lambda_i) - \sum_{\substack{i\in I^*\\ j\notin I^*}}\sum_{\substack{S\subset R_i\cap R_j, S\neq\{s_i\}\\ S\cap S^*=\emptyset}}\sum_{\substack{u\in S\\ v\in S^*\cap R_i}}\frac{d}{dt}\phi^i_{S,(uv)}\\
&= -\sum_{i\in I^*}(c_{s_i}-\lambda_i) - \sum_{\substack{i\in I^*\\ j\notin I^*}}\sum_{u\in R_i\cap R_j} c_u\\
&= -\sum_{i\in I^*}(c_{s_i}-\lambda_i) - \sum_{i\notin I^*}\sum_{u\in R_i\setminus\widetilde{R}_i} c_u
\end{aligned}$$

where the last equality follows since

$$\{u\in R_i\setminus\widetilde{R}_i : i\notin I^*\} = \{u\in R_i\cap R_j : i\in I^*, j\notin I^*\}.$$

Now we tackle the second term (B). Using a similar method to the single-commodity case, by counting the factor that each $\frac{d}{dt}\phi^i_{S,(uv)}$ term appears with, write

$$\begin{aligned}
(B) &= \sum_{i\notin I^*}\lambda_i(|R_i|-1) - \sum_{i\notin I^*}\sum_{S\subset R_i}\sum_{\substack{u\in S\cap\widetilde{R}_i\\ v\notin S, v\in R_i}}\frac{d}{dt}\phi^i_{S,(uv)}\\
&\leq \sum_{i\notin I^*}\lambda_i(|R_i|-1) - \sum_{i\notin I^*}\sum_{u\in\widetilde{R}_i} c_u\\
&\leq \sum_{i\notin I^*}\lambda_i\left(|R_{I\setminus I^*}| - |I\setminus I^*|\right) - \sum_{i\notin I^*}\sum_{u\in\widetilde{R}_i} c_u
\end{aligned}$$

where the summation in the first line follows since $\forall u\in R_i\setminus\widetilde{R}_i, v\in\widetilde{R}_i, S\subset R_i$, we have $\frac{d}{dt}\phi^i_{S,(uv)}=0$ since u is devoting all its capacity towards some most deprived neighbour in S^*, and away from nodes in R_i. thus, the only flow is from $\tilde{R}_i$ to R_i.

The reasoning for the last line is the following. For any $J\subseteq I$, we have

$$\begin{aligned}
|R_J| - |J| &\geq \max_{j\in J}|R_j| + |J| - 1 - |J|\\
&= \max_{j\in J}|R_j| - 1
\end{aligned}$$

since for each receiver set R_j, its source is contained in no other receiver set.

Combining (A) and (B) gives

$$\begin{aligned}(A)+(B) &\leq \sum_{i\in I^*}(\lambda_i - c_{s_i}) + \sum_{i\in I^*}\lambda_i(|R_{I\setminus I^*}| - |I\setminus I^*|) \\ &\quad - \sum_{i\notin I^*}\left(\sum_{u\in R_i} c_u + \sum_{u\in R_i\setminus \widetilde{R}_i} c_u\right) \\ &= \sum_{i\in I^*}(\lambda_i - c_{s_i}) + \sum_{i\in I^*}\lambda_i(|R_{I\setminus I^*}| - |I\setminus I^*|) - \sum_{u\in R_{I\setminus I^*}} c_u \\ &< 0\end{aligned}$$

under the condition (12). This proves Lemma 2. □

Proof of Theorem 3. Theorem 3 can be proved by combining Lemma 2 with Theorem 2. □

6 Neighbourhood Resampling

In this section we describe a random resampling procedure that allows us to use low degree neighbourhoods while maintaining optimal throughput. The technique we describe is reminiscent of the following randomized bipartite maximal matching algorithm of Tassiulas [7]: at time t choose a matching M uar from the set of possible matchings and compare its weight with the previous matching R_{t-1}; if it is better, set $R_t = M$, otherwise set $R_t = R_{t-1}$. Tassiulas proved that as long as the maximal matching has a nonzero probability of being chosen, then this procedure will give 100% throughput in an input-queued switch. With this simple example in mind, we can now describe our resampling procedure.

Recall the 'random-useful-to-most-deprived' (RMD) algorithm, with a single commodity and infinite download capacities for simplicity. Consider the following modification for fixed $d \geq 1, t \geq 0$. Each node u maintains a set of neighbours $N(u)$ of size d. After an exponentially distributed time with mean t, each node u chooses a new node v at random from the collection $\{v : (uv) \in E, v \notin N(u)\}$ and updates its neighbourhood by $N'(u) \leftarrow N(u)\cup\{v\}\setminus\{w\}$ iff $w \in N(u)$ satisfies $w = \arg\min_{z\in N(u)\cup\{v\}} X_{+u-z}$, i.e. v is a more deprived neighbour of u than some current neighbour w. At each time step, given the set $N(u)$ of selected neighbours, u performs RMD over them. We will show that, even for $d = 1$, this random resampling procedure

can give the same fluid limits as for the complete graph, i.e. when $N(u) = V$ at all time steps. We prove the following lemma.

Lemma 3 *For a node u let X_u be a random variable describing the number of time steps until y_{+u-v} is sampled. Consider any process P where there exists a constant κ such that $\Pr(X_u > \kappa) = \epsilon > 0$ for all nodes u. Then the fluid limits of RMD with P are equal to the fluid limits of RMD on the complete graph.*

Proof. For simplicity, we will consider only the case where there is a single commodity and the download capacity of nodes is infinite. Let $E(t)$ be the set of edges at time t, i.e. $\{(uv) : u \in V, v \in N_u(t)\}$. If $(uv) \in E(t)$ then we say that v is *selected* by u at t. We wish to establish that the fluid limits (3),(4) and (5) as established in Section 2 still hold. We shall consider the expanded state space $(\{X_S\}_{S\in\mathcal{S}}, A, E)$ that includes the current set of active edges.

Fix $h > 0$ and assume that u is such that $\sum_{v'\neq u} y_{+u-v'}(t) > 0$. If $y_{+u-v}(t) < \max_{v'\neq v} y_{+u-v}(t)$ then the same inequality holds for the interval $[t, t+h]$. Let $v^* \in \arg\max_{v'\neq u} y_{+u-v'}(t)$ be the most deprived node wrt u. Since $v \neq v^*$, the time until v is evicted is distributed geometrically. Hence there exists a constant $k > \kappa c_u$ such that the probability that u sends more than k packets to v in the interval $[Nt, N(t+h)]$ decreases exponentially with N (alternatively, for any $\epsilon > 0$ there exists a constant k' such that probability that the edge (uv^*) is inactive by $Nt + k'$ is less than ϵ). This means that

$$\lim_{N\to\infty} \frac{1}{h}\left(\frac{1}{N}\Phi^N_{S,(uv)}(N(t+h)) - \frac{1}{N}\Phi^N_{S,(uv)}(t)\right) = 0, \tag{18}$$

which establishes (3).

We shall assume that if u has no neighbours that it can give packets to, then it invokes the sampling procedure immediately. As above, there exists a constant k such that the probability that v^* is not selected by u by time $Nt + k$ decreases exponentially with N. Note that if v^* is selected at $Nt + k$ for $k \leq Nh$ then it shall remain selected until time $N(t+h)$. This implies that for large enough N we have the same equality as before, which implies that

$$\lim_{N\to\infty} \sum_{v\neq u, S\in\mathcal{S}: u\in S, v\notin S} \frac{1}{h}\left(\frac{1}{N}\Phi^N_{S,(uv)}(N(t+h)) - \frac{1}{N}\Phi^N_{S,(uv)}(Nt)\right) = c_u, \tag{19}$$

from which (4) follows.

Now assume that $u \neq s$ and v is such that $\sum_{S:u\in S, v\notin S} \frac{d}{dt}\phi_{S,(uv)}(t) > 0$, i.e. u is transmitting to v. Then of all instants during $[Nt, N(t+h)]$ when u sends a packet to v, a fraction $\frac{y_S(t)}{y_{+u-v}} + O(h + 1/N)$ of these are idle packets already replicated at S. Assume that $v \neq v^*$ (otherwise, there is no problem). We want to make sure that we transmit to a single node long enough so that the lower order term $O(h+1/N)$ disappears and we obtain the desired fractions. As before, choose a constant k such that the probability that v^* is not selected by u by time $Nt+k$ decreases exponentially with N. Once v^* is selected, it shall remain selected until time $N(t+h)$. Thus, letting $N \to \infty$ and for $h \to 0$ we find that u transmits to v^* for an arbitrarily large fraction of the interval $[Nt, N(t+h)]$. This establishes (5).

The case that $u = s$ is handled similarly to before, after accounting for the fact that we may waste a small amount of time transmitting to a node that is not the most deprived, but this can be made an arbitrarily small fraction of the time interval $[Nt, N(t+h)]$. This completes the proof of the lemma. □

It would be good to study the effect of the degree on the performance of the algorithm. In particular, it seems like using degree $d \geq 2$ will give substantially better results for convergence in establishing (19) since we can avoid the case that a node spends time searching for nodes to transmit to, if it only has a few possible neighbours with nonzero y_{+u-v} values, by scanning in the background.

6.1 Efficient resampling using expanders

We now describe how the resampling procedure can be efficiently implemented in a distributed setting.

Let H be a d-regular expander graph on n vertices. For sufficiently large n, one can obtain a uniform sample of size m from the space $\{1, \ldots, n\}$ by the following procedure: choose an intial vertex v_0 uar from H, then make a random walk $v_1, v_2, \ldots, v_{m-1}$ among the vertices of H, at each step choosing v_{i+1} uar from the neighbours of v_i.

For resampling, each node u does the following. Make a random sample $v_0, \ldots, v_{d-1}$ of size d as described above, and add all d vertices to the neighbourhood set $N(u)$. At each resampling instant t, let v_t be the next vertex chosen in the random walk on H and let $v_{\min} = \arg\min_{v_i \in N(u)} P_{+u-v_i}$. If $P_{+u-v_t} > P_{+u-v_{\min}}$ then update $N(u) \leftarrow (N(u) \setminus \{v_{\min}\}) \cup \{v_t\}$, otherwise $N(u)$ is unchanged.

7 Heavy traffic analysis

Consider the single-commodity case where $c_s = \lambda$ and $\sum_{u \in V} c_u > (n-1)\lambda$. In this situation, the previous stability results do not apply.

8 Freshness

In this section we prove the following result.

Theorem 4 *Assume that the graph $G = (V, E)$ is complete, and that $c_u \geq \lambda + \varepsilon$ for all nodes u and some $\varepsilon > 0$. Then the random useful forwarding algorithm is stable, even if the source does not prioritise fresh packets.*

Proof. The proof parallels that of Theorem 6 in [3], except for the case when $y_{\{s\}}(t) > 0$ and there exists a set S^* such that all nodes in S^* are deprived. We shall show that in this case, the work function $w(y(t))$ is still decreasing.

We shall use the fact that since $S \cap S^* = \emptyset$ we have $|S| \leq n - |S^*|$, hence $n - |S| \geq |S^*|$. Using this, we have

$$\begin{aligned}
\frac{d}{dt}w(y) &= \sum_{S \in \mathcal{S}, S \cap S^* = \emptyset} \frac{d}{dt} y_S(t)(n - |S|) \\
&\leq (n-1)\lambda - |S^*| \sum_{u \in S, S \cap S^* = \emptyset} \sum_{u \in S, v \in S^*} \frac{d}{dt}\phi_{S,(uv)}(t) \\
&= (n-1)\lambda - |S^*| \sum_{u \in S, S \cap S^* = \emptyset} c_u \\
&\leq (n-1)\lambda - |S^*|(n - |S^*|)(\lambda + \varepsilon) \\
&< 0.
\end{aligned}$$

The remainder of the proof follows Theorems 6 in [3] and is omitted. □

9 Open Problems

In this work we take another step towards advocating simple, local-control algorithms for network flow problems that previously relied on centralized algorithms for obtaining optimal solutions. There are some questions that arise from this work. Firstly, can we obtain explicit backlog and packet delay distributions? The rescaling techniques used in obtaining fluid limits hide

these explicit descriptions. Secondly, we would like to extend the protocol with a method for regulating the source injection rates to enforce some fairness properties when the set of feasible demands are unknown.

References

[1] J. Edmonds. Edge-disjoint branchings. In R. Rustin, editor, *Combinatorial Algorithms*, pages 21–31. Algorithmics Press, 1972.

[2] Lestor R. Ford, Jr., and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[3] L. Massoulie, A. Twigg, C. Gkantsidis, and P. Rodriguez. Provably optimal decentralized broadcast algorithms. Technical Report MSR-TR-2006-105, Microsoft Research, Jul 2006.

[4] Laurent Massoulié, Andrew Twigg, Christos Gkantsidis, and Pablo Rodriguez. Decentralized broadcasting algorithms. In *Proceedings of Infocom 2007*, 2007.

[5] Karl Menger. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927.

[6] P. Robert. *Stochastic Networks and Queues*. Springer, 2003.

[7] Leandros Tassiulas. Linear complexity algorithms for maximum througput in radio networks and input queued switches. In *INFOCOM*, pages 533–539, 1998.