

Sending Perishable Information: Coding Improves Delay-Constrained Throughput Even for Single Unicast

Chih-Chun Wang

School of Electrical and Computer Engineering
Purdue University, USA

Minghua Chen

Department of Information Engineering
The Chinese University of Hong Kong, Hong Kong

Abstract—We consider a delay-constrained unicast scenario, where a source node streams perishable information to a destination node over a directed acyclic graph subject to a delay constraint. Transmission along any edge incurs unit delay, and we require that every information bit generated at the source in the *beginning* of time t to be received and recovered by the destination in the *end* of time $t + D - 1$ where $D > 0$ is the maximum allowed communication delay. We study the corresponding *delay-constrained (d-cn) unicast capacity problem*.

When only routing is allowed, [Ying, *et al.* 2011] showed that the aforementioned d-cn unicast routing capacity can be characterized and computed efficiently. However, the d-cn capacity problem changes completely when network coding (NC) is allowed. In this work, we construct the first example showing that NC can achieve strictly higher d-cn throughput than routing even for the single unicast setting and the NC gain can be arbitrarily close to 2 in some instances. This is in sharp contrast to the delay-unconstrained ($D \rightarrow \infty$) single-unicast case where the classic min-cut/max-flow theorem implies that coding cannot improve throughput over routing. Finally, we propose a new upper bound on the d-cn unicast NC capacity and elaborate its connections to the existing routing-based results [Ying, *et al.* 2011]. Overall, our results suggest that d-cn communication is fundamentally different from the well-understood delay-unconstrained one and call for investigation participation.

I. INTRODUCTION

Consider a network modeled as a directed acyclic graph G , for which each edge has a capacity constraint and incurs a unit transmission delay. We consider exclusively a delay-constrained (d-cn) single-unicast scenario where a single source node, denoted as s , streams perishable information to a single destination node, denoted as d , over the graph G . Every information bit generated at s in the *beginning* of time t has to be received and recovered by d by the *end* of time $t + D - 1$. Namely, the maximum allowed end-to-end communication delay of any packet is $(t + D - 1) - t + 1 = D$, where the value of D is specified by the delay requirement of the applications.

In this paper, we study the *d-cn unicast capacity problem*, *i.e.*, computing and achieving the maximum rate at which s can stream perishable information to d subject to the delay constraint D .

The problem is important for delay-sensitive multimedia communication systems, and for delivering real-time control messages for cyber-physical systems. In general, an optimal d-cn communication scheme needs to decide the optimal routes of the information flow *in space* in order to fully utilize

all the link capacity resources, while simultaneously tracking the delay of individual packets *in time* to ensure the packets can arrive at d and the information can be recovered before expiration. The design problem becomes even more involved when we allow for *network coding (NC)* [1] at intermediate nodes that intelligently mix the information content in packets before forwarding them. Such a 3-way coupling among space, time, and NC choices creates a unique challenge and our understanding of d-cn network capacity is still nascent.

When D is sufficiently large (*e.g.*, larger than the end-to-end delay of the longest path between s and d), any communication scheme can always meet the delay constraint. Therefore, the delay-unconstrained (since $D \rightarrow \infty$) single-unicast capacity can be characterized by the classic min-cut/max-flow theorem, and an optimal routing solution can be obtained in polynomial time using the Ford-Fulkerson algorithm [2]. Since optimal routing already achieves the capacity, *i.e.*, the min-cut value, NC cannot improve throughput over optimal routing when there is only one unicast flow in the network.

The story changes completely when D is small (*i.e.*, when the delay constraint is active). For example, the d-cn unicast routing capacity has to be computed by the concept of *soft edge-cuts* [3], [4], which is different from the standard graph-theoretic notion of edge cuts. Also, as will be illustrated in Section III, there are some simple network instances for which optimal routing can achieve strictly higher d-cn throughput than random linear network coding (RLNC), a sharp contrast to the delay-unconstrained case in which both RLNC and optimal routing can achieve the single-unicast capacity [5].

Overall, we observe that the landscape of d-cn unicast is fundamentally different from the well-understood delay-unconstrained one. In this paper, we study the d-cn unicast capacity problem and make the following contributions.

▷ This work shows for the first time in the literature that for d-cn traffic, NC can achieve strictly higher throughput than optimal routing *even for single unicast* and the NC gain be arbitrarily close to 2.

The result is interesting in the following sense. Most of the Internet traffic is unicast. One of the fundamental results in NC is that routing achieves the single-unicast capacity when there is no delay constraint. This implies that to capitalize the NC benefits for delay-insensitive unicast traffic, one has to perform NC over multiple coexisting flows, the so-called inter-flow NC. It is known that designing the optimal inter-

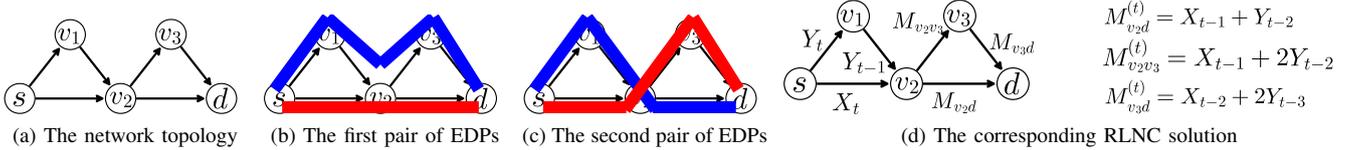


Fig. 1: A simple example with $D = 3$ that demonstrates the key differences of the delay-constrained setting.

flow NC scheme is a notoriously hard problem, see [6] and the references therein. What exacerbates the problem is that even if we can design an optimal inter-flow NC scheme in a theoretic setting, in practice inter-flow NC requires additional coordinations among participating flows, including the tasks of hand-shaking, synchronization, joint buffer management, *etc.* Our result suggests that one may use NC to improve the performance of delay-sensitive traffic over optimal routing without any coordination among coexisting network flows!

▷ We also propose a new upper bound on the d-cn unicast NC capacity, which provides deeper understanding to the overall d-cn network communication problem and sheds further insights to the existing routing-based d-cn results in [3].

II. COMPARISON TO EXISTING WORKS

Decoding delay of NC is a very well studied problem, see [7], [8] and the references therein. Almost all existing works focus on how to minimize the decoding delay when using NC to attain the best possible throughput for delay-insensitive traffic. Namely, attaining the absolute optimal throughput is of the highest priority, and minimizing the delay is to ensure that the extra (decoding) delay incurred by NC is not excessive. Many of the existing results also focus on 1-hop networks with *random packet erasure*.

In contrast with the existing throughput-centric delay studies [7], [8], this work is delay-centric. Namely, we consider a hard delay constraint such that any packets that experience delay longer than D time slots are deemed useless and discarded. With the highest priority being the delay constraint, we study how to maximize the throughput over any given *error-free multi-hop network*. This approach is a significant departure from the existing works on minimizing NC decoding delay.

Remark: This work focuses exclusively on the multi-hop environment. However, if we further restrict our focus to the 1-hop setting, the concept of hard delay constraints is related to the completion time analysis of index coding [9].

III. A SIMPLE EXAMPLE

Consider the example in Fig. 1(a). The min-cut value between s and d is 2, which implies the existence of (at least) one pair of edge-disjoint paths (EDPs). There are actually two possible pairs of EDPs, see Figs. 1(b) and 1(c), respectively. Assume each edge incurs a unit delay. If there is no delay constraint, we can sustain throughput 2 by *routing* the packets either through Fig. 1(b) or through Fig. 1(c). However, with delay constraint $D = 3$, only the two paths in Fig. 1(c) can be used to transmit information at rate 2. For comparison, one

path in Fig. 1(b) has 4 hops, and the information transmitted along that path will expire before arriving at d .

We now apply RLNC to Fig. 1(a) while assuming a sufficiently large finite field $\text{GF}(q)$ is used, say $\text{GF}(3)$. In the beginning of each time t , we send packets $X_t \in \text{GF}(q)$ and $Y_t \in \text{GF}(q)$ along the edges (s, v_2) and (s, v_1) , respectively. Due to the unit-delay incurred in edge (s, v_1) , in the beginning of time t , we can send Y_{t-1} along (v_1, v_2) . Node v_2 can now perform RLNC. We can assume, without loss of generality, that in the beginning of time t node v_2 sends $M_{v_2d}^{(t)} = X_{t-1} + Y_{t-2}$ and $M_{v_2v_3}^{(t)} = X_{t-1} + 2Y_{t-2}$ along (v_2, d) and (v_2, v_3) , respectively. Following similar derivations, by the *end* of time t , destination d should have received $M_{v_3d}^{(t)} = X_{t-2} + 2Y_{t-3}$ and $M_{v_2d}^{(t)} = X_{t-1} + Y_{t-2}$.

Since s starts to send Y_t and X_t in the beginning of time t for all $t \geq 1$, we set $X_t = Y_t = 0$ for all $t \leq 0$. From the above derivations, by the “end” of time 3, d has received $M_{v_3d}^{(3)} = X_1 + 2Y_0 = X_1$ and $M_{v_2d}^{(3)} = X_2 + Y_1$. Recall that $D = 3$. Therefore, d is interested in decoding both X_1 and Y_1 , which were sent by s in the “beginning” of time 1 (3 slots earlier). One can verify that *knowing* $M_{v_3d}^{(3)} = X_1$ and $M_{v_2d}^{(3)} = X_2 + Y_1$ is *not sufficient* for d to decode the desired X_1 and Y_1 since the value of Y_1 in $M_{v_2d}^{(3)}$ is now “corrupted” by the future packet X_2 that has not been decoded yet. Actually, even when time progresses, d is not able to decode both X_t and Y_t by the end of time slot $(t+D-1) = t+2$ for any $t > 1$. One can prove that the RLNC throughput of this example is 1, which is strictly less than the routing capacity 2.

IV. RESULTS WHEN ONLY ROUTING IS ALLOWED

We model the network as a finite directed acyclic graph $G = (V, E)$, where V is the node set and E is the edge set. We use $\text{In}(v)$ and $\text{Out}(v)$ to denote the collections of the incoming and outgoing edges of v , respectively. For any $e = (u, v) \in E$, we define $\text{tail}(e) \triangleq u$ and $\text{head}(e) \triangleq v$. Each e has a capacity constraint c_e and incurs unit delay. Links with long delay are thus modeled as a path of multiple edges. With delay constraint D , any packet traverses from s to d through a path longer than D hops is deemed useless. Without loss of generality, we assume $D \leq |E|$. Otherwise, the problem collapses back to the classic delay-unconstrained problem since all paths have length $\leq |E|$. We also assume $\text{In}(s) = \emptyset$ and $\text{Out}(d) = \emptyset$. For any integer k , we define $[1, k] \triangleq \{1, 2, \dots, k\}$ and define $[1, \infty)$ as the set of positive integers.

Let \mathcal{P}_D denote the collection of all s -to- d paths of length “ $\leq D$ hops.” Obviously \mathcal{P}_D is finite. The largest d-cn routing

capacity, denoted by R_{route}^* , can be computed by the following LP problem:¹

$$\max_{\{x_P \geq 0: P \in \mathcal{P}_D\}} \sum_{P \in \mathcal{P}_D} x_P \quad (1)$$

$$\text{subject to } \forall e \in E, \sum_{P: P \ni e, P \in \mathcal{P}_D} x_P \leq c_e, \quad (2)$$

which consists of $|E|$ inequalities and $|\mathcal{P}_D|$ non-negative variables $\{x_P\}$, each x_P representing the rate along path P . The objective (1) is the sum of the throughput sent over the $|\mathcal{P}_D|$ paths, and (2) imposes that the sum rate of all paths using an edge e should not exceed c_e . However, since $|\mathcal{P}_D|$ grows exponentially with respect to $|G| \triangleq |V| + |E|$, the above LP characterization is not easily computable for large networks. To address the above concern of complexity, [3] states the following result.

Proposition 1 (Sec. IV.A [3]): We can compute R_{route}^* by the following flow-based LP problem with $|E| \cdot D$ non-negative variables $x_e^{(h)}$ representing the total rate of flows traveling their h -th hop on link e for all $e \in E$ and $h \in [1, D]$:

$$\max_{x_e^{(h)} \geq 0} \sum_{e \in \text{In}(d)} \sum_{h=1}^D x_e^{(h)} \quad (3)$$

$$\text{subject to } \forall v \in V \setminus \{s, d\}, \forall h \in [1, D], \sum_{e \in \text{In}(v)} x_e^{(h-1)} = \sum_{\tilde{e} \in \text{Out}(v)} x_{\tilde{e}}^{(h)} \quad (4)$$

$$\forall e \in E, \sum_{h=1}^D x_e^{(h)} \leq c_e. \quad (5)$$

Here the objective in (3) is the aggregate rate of flows that arrive at d within D hops. The constraints in (4) say that the aggregate incoming flows to node v with hop count $h-1$ must be equal to the aggregate outgoing flows from node v with hop count h ; these are essentially the flow balance equations with flow travelled-distance (in hops) taken into account. The constraints in (5) are link capacity constraints. Note that in (4) we use the convention that $x_e^{(0)} = 0$ for all $e \in E$.

Since the proof of Proposition 1 was omitted in [3], we sketch the proof in the following for completeness.

Proof: We first prove that any solution in the LP problem (1)–(2) leads to a valid solution for the LP problem (3)–(5). This is done by setting

$$x_e^{(h)} = \sum_{P: \text{the } h\text{-th hop of path } P \text{ is } e} x_P.$$

With the above construction of $x_e^{(h)}$, one can see that (4) holds naturally and (1) equals to (3). Also, (2) on x_P implies (5). The forward direction is thus proven.

We now prove that any solution in (3)–(5) leads to a valid solution in (1)–(2). We prove this by an iterative construction. Initially, we define $\underline{\mathcal{P}} = \mathcal{P}_D$ and $\underline{x}_e^{(h)} = x_e^{(h)}$ for all e and h . For any $P \in \underline{\mathcal{P}}$, we choose $x_P = \min_{h \in [1, |P|]} \underline{x}_{e_{P,h}}^{(h)}$ where

$e_{P,h}$ is the h -th edge of P . After choosing x_P , we decrease the value of $\underline{x}_{e_{P,h}}^{(h)}$ by x_P for all $h \in [1, |P|]$ and remove P from $\underline{\mathcal{P}}$. After decreasing the $\underline{x}_{e_{P,h}}^{(h)}$ values and reducing $\underline{\mathcal{P}}$, we repeat the construction for another $\hat{P} \in \underline{\mathcal{P}}$ until $\underline{\mathcal{P}} = \emptyset$.

Claim 1: Throughout the process, all the $\underline{x}_e^{(h)}$ are non-negative and they satisfy (4). The non-negativity holds since $x_P = \min_{h \in [1, |P|]} \underline{x}_{e_{P,h}}^{(h)}$. Equality (4) holds since we subtract x_P from all $\underline{x}_{e_{P,h}}^{(h)}$. Note that $\underline{x}_e^{(h)} \geq 0$ also implies that the resulting x_P is non-negative. *Claim 2:* The resulting x_P satisfies (2). To prove this claim, we notice that since x_P is deducted from $\underline{x}_{e_{P,h}}^{(h)}$ during our construction, we always have

$$\forall e \in E, \sum_{P: P \ni e, P \in \mathcal{P}_D \setminus \underline{\mathcal{P}}} x_P + \sum_{h=1}^D \underline{x}_e^{(h)} = \sum_{h=1}^D x_e^{(h)} \quad (6)$$

in the end of each iteration. Then by (5) and by the non-negativity of $\underline{x}_e^{(h)}$, the resulting x_P in the end (i.e., $\underline{\mathcal{P}} = \emptyset$) must satisfy (2). *Claim 3:* The expression (1) computed from the final x_P equals (3) computed from $x_e^{(h)}$. To prove this claim, we notice that by the construction of \mathcal{P}_D , the last edge of any $P \in \mathcal{P}_D$ must belong to $\text{In}(d)$. Since (6) holds for any edge $e \in \text{In}(d)$, we only need to prove that $\underline{x}_e^{(h)} = 0$ for all $e \in \text{In}(d)$ and $h \in [1, D]$ in the end of our construction.

We prove this by contradiction. Suppose not. Then we have $\underline{x}_{\hat{e}}^{(\hat{h})} > 0$ for some $\hat{e} \in \text{In}(d)$ and \hat{h} . Since $\{\underline{x}_e^{(h)}\}$ satisfies (4) for all $v \in V \setminus \{s, d\}$ and h , we can find \hat{h} edges, denoted by \hat{e}_1 to $\hat{e}_{\hat{h}}$, satisfying simultaneously (i) $\hat{e}_{\hat{h}} = \hat{e}$; (ii) $\hat{e}_1 \hat{e}_2 \cdots \hat{e}_{\hat{h}}$ form a path of length \hat{h} , which is denoted by \hat{P} ; and (iii) $\underline{x}_{\hat{e}_i}^{(i)} > 0$ for $i \in [1, \hat{h}]$. Since $\hat{e}_{\hat{h}} = \hat{e} \in \text{In}(d)$, we have $\text{head}(\hat{e}_{\hat{h}}) = d$. We now prove $\text{tail}(\hat{e}_1) = s$ by contradiction. Suppose not. Then we focus on (4) with $h = 1$ and $v = \text{tail}(\hat{e}_1)$. One can see that the left-hand side of (4) is zero since $\underline{x}_e^{(0)} = 0$ in our convention but the right-hand side is no less than $\underline{x}_{\hat{e}_1}^{(1)} > 0$. This contradiction implies that $\text{tail}(\hat{e}_1) = s$. As a result, \hat{P} connects s and d using $\hat{h} \leq D$ hops. Therefore $\hat{P} \in \mathcal{P}_D$.

On the other hand, all the paths in \mathcal{P}_D must have been considered in the iterative construction. Consequently, for any path $P \in \mathcal{P}_D$, $\underline{x}_{e_{P,h}}^{(h)} = 0$ for at least one $h \in [1, |P|]$ since we subtract $x_P = \min_{h \in [1, |P|]} \underline{x}_{e_{P,h}}^{(h)}$ from all $\underline{x}_{e_{P,h}}^{(h)}$. Property (iii) of $\hat{P} \in \mathcal{P}_D$ thus contradicts the fact that we have exhaustively considered all $P \in \mathcal{P}_D$. Claim 3 is thus proven.

Jointly, Claims 1 to 3 complete the proof. \blacksquare

By converting (1)–(2) to its dual problem, R_{route}^* can also be computed by the following cut-based LP problem.

$$\min_{\{y_e \geq 0: e \in E\}} \sum_{e \in E} y_e c_e \quad (7)$$

$$\text{subject to } \forall P \in \mathcal{P}_D, \sum_{e: e \in P} y_e \geq 1. \quad (8)$$

Note that if we replace \mathcal{P}_D by \mathcal{P}_∞ , the latter of which contains all paths regardless of their lengths, then one can prove that for any given network instance, (one of) the minimizing $\{y_e^*\}$ of (7)–(8) satisfies $y_e^* \in \{0, 1\}, \forall e \in E$. Solving (7)–(8) is no different than finding the minimum edge

¹It is sometimes called the hop-count-constrained max-flow problem.

cut (those e with $y_e^* = 1$). However, with \mathcal{P}_D , the minimizing $\{y_e^*\}$ can sometimes be fractional. Therefore, R_{route}^* is now characterized by some kind of *soft* min-cut, a unique feature of the d-cn setting. Examples of y_e^* being fractional can be found in the end of Section V-A and in [4].

Also observe that there are $|\mathcal{P}_D|$ inequalities in (8), which grows exponentially with respect to $|G|$. In contrast, Proposition 1 has the following easily computable dual problem.

Corollary 1: We can compute R_{route}^* by the following LP problem with $|E|$ non-negative variables $y_e \geq 0, \forall e \in E$, and $(|V| - 2) \cdot D$ real-valued variables $y_v^{(h)}, \forall v \in V \setminus \{s, d\}, h \in [1, D]$ such that

$$\begin{aligned} \min_{y_e \geq 0, y_v^{(h)}} \quad & \sum_{e \in E} y_e c_e & (9) \\ \text{s.t.} \quad & y_e + y_{\text{head}(e)}^{(h+1)} - y_{\text{tail}(e)}^{(h)} \geq 0, \forall e \in E, \forall h \in [1, D], & (10) \end{aligned}$$

where in (10) we use the convention $y_v^{(D+1)} = 0$ for all $v \in V$ and $y_s^{(h)} = y_d^{(h)} = 0$ for all $h \in [1, D]$.

V. NC \succ ROUTING EVEN FOR SINGLE-UNICAST

Without delay constraint, one fundamental result of NC is

$$R_{\text{NC}}^* = R_{\text{route}}^* \quad (11)$$

for any single-unicast flow from s to d , where R_{NC}^* is the NC capacity. We have found that (11) no longer holds for d-cn traffic (*i.e.*, when D is finite and small).

A. A Simple Network Example with NC gain = $\frac{4}{3}$

Consider the network in Fig. 2 with all edges having $c_e = 1$ and the delay constraint $D = 6$. We will show that such a network has $R_{\text{NC}}^* = 2 > R_{\text{route}}^* = 1.5$.

We first prove that $R_{\text{NC}}^* = 2$ by explicit NC construction. In Fig. 2, two packets X_t and Y_t are sent by s in the beginning of time t . Since each edge incurs unit delay, we have $M_{v_1 v_2}^{(t)} = X_{t-1}$ and $M_{v_6 v_2}^{(t)} = Y_{t-2}$ along edges (v_1, v_2) and (v_6, v_2) , respectively. We then let v_2 mix the two incoming packets and send $M_{v_2 v_3}^{(t)} = X_{t-2} + Y_{t-3}$ along (v_2, v_3) . By accounting for the delay incurred along the paths, we have $M_{v_3 v_4}^{(t)} = X_{t-3} + Y_{t-4}$, $M_{v_{10} v_4}^{(t)} = X_{t-3}$, and $M_{v_8 d}^{(t)} = X_{t-5} + Y_{t-6}$. Fig. 2 contains the summary of our NC choices thus far except for the $M_{v_4 d}^{(t)}$ message. The remaining question to be answered is what is the right NC choice at node v_4 ?

If we perform RLNC at v_4 , then v_4 will simply mix the two incoming packets together and send

$$\text{RLNC: } M_{v_4 d}^{(t)} = M_{v_3 v_4}^{(t-1)} + M_{v_{10} v_4}^{(t-1)} = 2X_{t-4} + Y_{t-5}. \quad (12)$$

Recall that $D = 6$. One can then verify that in the end of time 6, node d receives $M_{v_8 d}^{(6)} = X_1 + Y_0 = X_1$ and $M_{v_4 d}^{(6)} = 2X_2 + Y_1$ since we assume $X_t = Y_t = 0$ for all $t \leq 0$. Similar to the example discussed in Section III, d cannot decode Y_1 since Y_1 is corrupted by X_2 , which has not been decoded yet.

On the other hand, we can perform the following optimal NC instead. That is, instead of “adding” the two incoming

packets, we now “subtract”² $M_{v_{10} v_4}^{(t-1)}$ from $M_{v_3 v_4}^{(t-1)}$:

$$\text{Optimal: } M_{v_4 d}^{(t)} = M_{v_3 v_4}^{(t-1)} - M_{v_{10} v_4}^{(t-1)} = Y_{t-5}. \quad (13)$$

Node d can now decode X_1 and Y_1 from $M_{v_8 d}^{(6)} = X_1$ and $M_{v_4 d}^{(6)} = Y_1$ within the delay constraint. An astute reader may notice that in the end of time 7, d has received $M_{v_8 d}^{(7)} = X_2 + Y_1$ and $M_{v_4 d}^{(7)} = Y_2$, where X_2 in $M_{v_8 d}^{(7)}$ is “corrupted” by Y_1 . *Nonetheless, d can remove Y_1 in the end of time 7 since d has decoded Y_1 in the end of time 6.* The above argument can be used to prove that d can decode X_t and Y_t (injected in the beginning of time t) by the end of time $t + 5, \forall t \geq 1$. The $D = 6$ constraint is met. Since $\text{min-cut}(s, d) = 2$ in Fig. 2, we have d-cn NC capacity being $R_{\text{NC}}^* = 2$ packets per slot.

We then apply (7)–(8) to Fig. 2 and derive $R_{\text{route}}^* = 1.5$. The corresponding minimizing y_e^* are: $y_{s v_1}^* = y_{v_2 v_3}^* = y_{v_4 d}^* = 0.5$ and all other $y_e^* = 0$. This example shows that NC strictly outperforms optimal routing even for the single-unicast setting!

B. How Large Can The NC Gain Be?

In the previous example, the NC throughput gain over routing is $\frac{R_{\text{NC}}^*}{R_{\text{route}}^*} = \frac{2}{1.5}$. An interesting open question is *what is the largest NC gain in a single-unicast d-cn setting?* Specifically, we are interested in quantifying

$$\sup_{G \in \mathcal{G}_{s-u}, D \in [1, \infty)} \text{gain}_{s-u}(G, D) \quad (14)$$

where \mathcal{G}_{s-u} contains all possible network instances with single-unicast (s-u) traffic, and $\text{gain}_{s-u}(G, D)$ is the single-unicast NC gain over routing in G with delay constraint D .

One can easily prove that *the d-cn NC gain can be unbounded for the single-multicast (s-m) networks and for the multiple-unicast (m-u) networks*, denoted by \mathcal{G}_{s-m} and \mathcal{G}_{m-u} , respectively. Namely,

$$\sup_{G \in \mathcal{G}_{s-m}, \forall D} \text{gain}_{s-m}(G, D) \geq \sup_{G \in \mathcal{G}_{s-m}} \text{gain}_{s-m}(G, 2) = \infty \quad (15)$$

$$\sup_{G \in \mathcal{G}_{m-u}, \forall D} \text{gain}_{m-u}(G, D) \geq \sup_{G \in \mathcal{G}_{m-u}} \text{gain}_{m-u}(G, 3) = \infty \quad (16)$$

where the equality in (15) follows from the *combination network* construction in [10] and the equality in (16) follows from the *extended butterfly* construction in [11].

Nonetheless, the proofs of (15) and (16) cannot be applied to the single-unicast setting since they rely heavily on the fact that there are multiple destinations so that different destinations can either capitalize the diversity gain (for single multicast) or smartly cancel the interference of the other coexisting flows (for multiple unicast). These types of gains do not exist when there is only one destination³ in the network!

Our best understanding of (14) is summarized as follows and the corresponding proof is omitted due to the page limit.

²To distinguish between adding and subtracting, we assume GF(3) is used.

³In the single-unicast setting, one needs to consider a different type of interference. That is, optimal NC needs to remove the corruption caused by *future, not-yet decoded packets* within the same flow. See the detailed discussion of the suboptimal RLNC choice (12) versus the optimal choice (13). Such a new notion of interference is strongly coupled with the time-axis and calls for the development of new analysis tools.

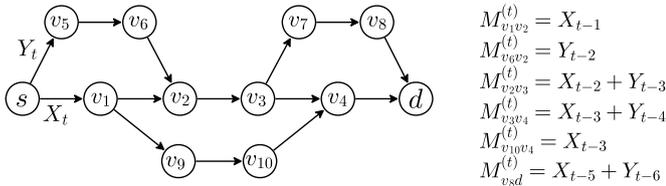


Fig. 2: A simple network with NC gain $\triangleq \frac{R_{\text{NC}}^*}{R_{\text{route}}^*} = \frac{4}{3}$.

Proposition 2: For any $0 < \epsilon < 2$, there exists a network $G \in \mathcal{G}_{s-u}$ and delay constraint D satisfying

$$\frac{R_{\text{NC}}^*}{2 - \epsilon} \geq R_{\text{route}}^* \geq 1, \text{ which implies } \frac{R_{\text{NC}}^*}{R_{\text{route}}^*} \geq 2 - \epsilon.$$

In a broad sense, R_{route}^* characterizes the maximum number of EDPs with length $\leq D$ hops, along which we can “squeeze through” R_{route}^* packets before expiration. Therefore, at least heuristically, any additional packets sent over the network (other than the original R_{route}^* packets) are either dependent or experiencing too long delay. Proposition 2 implies a rather counter-intuitive result: *With carefully-designed NC, those additional “useless” packets (either dependent or experiencing too long delay) can help us double the number of independent packets that can be decoded by d within the delay constraint.*

C. Upper Bounding The NC Capacity

The d-cn R_{route}^* naturally serves as a lower bound on R_{NC}^* . We now present an upper bound on R_{NC}^* .

Proposition 3: The following integer programming problem computes an upper bound UB_{NC} on R_{NC}^* :

$$\min_{\{y_e: e \in E\}} \sum_{e \in E} y_e c_e \quad (17)$$

$$\text{subject to } \forall P \in \mathcal{P}_D, \sum_{e: e \in P} y_e \geq 1. \quad (18)$$

$$\forall e \in E, y_e \in \{0, 1\}. \quad (19)$$

The corresponding proof combines the *time-expanded network representation* [1] and the *generalized network-sharing bound* [12]. The details are omitted due to the page limit.

Note that another simple upper bound can be derived as follows. Take any given \mathcal{P}_D , we use $G|_{\mathcal{P}_D}$ to denote the subgraph induced by \mathcal{P}_D . Then by the standard cut-set bound arguments, the min-cut value separating (s, d) in $G|_{\mathcal{P}_D}$ is an upper bound on R_{NC}^* . However, such an upper bound is looser than Proposition 3. Let us reuse the example in Fig. 1(a) for illustration. We set $D = 3$, set $c_{sv_2} = c_{v_2d} = 1$, and set $c_e = 2$ for all other edges. The delay-respecting path-set \mathcal{P}_D contains 3 paths, i.e., sv_2d , sv_1v_2d , and sv_2v_3d , and we have $G = G|_{\mathcal{P}_D}$. The min-cut value separating s and d in $G|_{\mathcal{P}_D}$ is 3. Meanwhile, by setting $y_{sv_1} = y_{v_2d} = 1$ and all other $y_e = 0$, we satisfy (18)–(19) and obtain the upper bound $\text{UB}_{\text{NC}} = 2 \geq R_{\text{NC}}^*$, which is strictly stronger than the simple min-cut bound. Note that the two edges $\{(s, v_1), (v_2, d)\}$ break every path in \mathcal{P}_D but do not separate s from d in $G|_{\mathcal{P}_D}$; thus they do not form a cut in $G|_{\mathcal{P}_D}$ by standard definition.

Comparing Proposition 3 with (7) and (8), we see that *adding the integer condition (19) to the minimization problem turns R_{route}^* , a lower bound on R_{NC}^* , to an upper bound UB_{NC} on R_{NC}^* .* Proposition 3 thus implies that for any network instance, if the minimizing y_e^* of (7)–(8) is integral, then the lower and upper bounds match and we have fully characterized the d-cn unicast NC capacity: $R_{\text{route}}^* = R_{\text{NC}}^* = \text{UB}_{\text{NC}}$. On the other hand, for any network instance in which $R_{\text{NC}}^* > R_{\text{route}}^*$, e.g., Fig. 2, the y_e^* of (7)–(8) must be fractional, which was observed in the end of Section V-A.

VI. CONCLUSION

This work studies the following problem: Given a hard delay constraint, how much perishable information one can send from s to d . We have proven that NC can strictly outperform optimal routing even for the single-unicast setting and the gain can be arbitrarily close to 2. Although complete characterization of the delay-constrained NC capacity R_{NC}^* remains an open problem, we have identified a new upper bound on R_{NC}^* . Overall, our results suggest that delay-constrained communication is fundamentally different from the well-understood delay-unconstrained one and call for investigation participation.

ACKNOWLEDGEMENT

The work described in this paper was supported in part by NSF grant CCF-0845968, National Basic Research Program of China (Project No. 2012CB315904 and 2013CB336700), and two grants from the University Grants Committee of the Hong Kong Special Administrative Region, China (AoE Project No. AoE/E-02/08 and GRF Project No. 411011).

REFERENCES

- [1] S.-Y. Li, R. W. Yeung, and N. Cai, “Linear network coding,” *IEEE Trans. Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.
- [2] L. R. Ford and D. R. Fulkerson, “Maximal flow through a network,” *Canadian Journal of Mathematics*, vol. 8, no. 3, pp. 399–404, 1956.
- [3] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, “On combining shortest-path and back-pressure routing over multihop wireless networks,” *IEEE Trans. Networking*, vol. 19, no. 3, pp. 841–854, June 2011.
- [4] M. Kodialam and T. Lakshman, “On allocating capacity in networks with path length constrained routing,” in *Proc. Allerton*, 2002.
- [5] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [6] R. Dougherty, C. Freiling, and K. Zeger, “Networks, matroids, and non-Shannon information inequality,” *IEEE Trans. Inf. Theory*, vol. 53, no. 6, pp. 1949–1969, June 2007.
- [7] A. Eryilmaz, A. Ozdaglar, M. Médard, and E. Amed, “On the delay and throughput gains of coding in unreliable networks,” *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5511–5524, December 2008.
- [8] E. Drinea, C. Fragouli, and L. Keller, “Delay with network coding and feedback,” in *Proc. IEEE Int’l Symp. Inform. Theory*, 2009.
- [9] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, “Index coding with side information,” in *Proc. Annual IEEE Symp. Found. Computer Science*. Berkeley, California, USA, October 2006.
- [10] C. Ngai and R. Yeung, “Network coding gain of combination networks,” in *Proc. IEEE Inform. Theory Workshop*. San Antonio, Texas, October 2004, pp. 283–287.
- [11] N. Harvey, R. Kleinberg, and A. Lehman, “On the capacity of information network,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2345–2364, June 2006.
- [12] S. Kamath, D. Tse, and V. Anantharam, “Generalized network sharing outer bound and the two-unicast problem,” in *Proc. 7th Workshop on Network Coding, Theory, & Applications (NetCod)*. Beijing, China, July 2011, pp. 1–6.