Optimal Flow Distribution among Multiple Channels with Unknown Capacities

Richard Karp Till Nierhoff* Till Tantau*

International Computer Science Institute
1947 Center St Suite 600
Berkeley, CA 94704
{karp,nierhoff,tantau}@icsi.berkeley.edu

2nd July 2004

Abstract

Consider a simple network flow problem in which there are n channels directed from a source to a sink. The channel capacities are unknown and we wish to determine a feasible network flow of value D. Flow problems with unknown capacities arise naturally in numerous applications, including inter-domain traffic routing in the Internet, bandwidth allocation for sending files in peer-to-peer networks, but also the distribution of physical goods like newspapers among different points of sale. We study protocols that probe the network by attempting to send a flow of at most D units through the network. If the flow is not feasible, the protocol is told on which channels the capacity was exceeded (binary feedback) and possibly also how many units of flow were successfully send on these channels (throughput feedback). For the latter, more informative type of feedback we present optimal protocols for minimizing the number rounds needed to find a feasible flow and for minimizing the total amount of wasted flow. For binary feedback, we show that one can exploit the fact that network capacities are often larger than the demand D: We present a protocol for this situation that is optimal under certain assumptions and finds solution more quickly than the generalized binary search protocol previously proposed in the literature. For the special case of two channels we present a protocol that is optimal without any extra assumptions and outperforms binary search.

1 Introduction

Problem Statement. We study the network flow problem in which there are n channels directed from a source to a sink, and we wish to determine a feasible flow of value D from the source to the sink. Each channel i has a capacity c_i. Initially, these capacities are unknown, but we know that they are nonnegative, integral, and sum up to some $C \geq D$. We also assume that the capacities do not change over time, that is, we only consider the static case in the present paper. To determine a feasible flow we proceed in rounds $t = 1, 2, 3, \ldots, T$. In round t we choose a nonnegative *query vector* $q(t) = (q_1, q_2, \ldots, q_n)$, and simultaneously for all i, attempt to send q_i units of flow through channel i. The queries are nonnegative, rational, and sum up to at most D. We then receive feedback about the success of our attempts in the form of a *feedback vector* $f(t) = (f_1, \ldots, f_n)$. In the case of *binary feedback* we learn, for each channel i, whether all of our

*The authors were supported through DAAD (German academic exchange service) postdoc fellowships.

flow reached the sink; that is, $f_i = \mathit{success}$ when $q_i \le c_i$ and $f_i = \mathit{failure}$ otherwise. In the case of *throughput feedback*, we learn how much flow was delivered through each channel; that is, $f_i = \min\{q_i, c_i\}$.

We study the efficient choice of the successive query vectors. We may be interested in minimizing the number of rounds required to determine a feasible flow, or in finding a feasible flow with minimum total waste. In the latter case, the *throughput* $P(t) = \sum_{i=1}^{n} \min\{q_i, c_i\}$ of round t is the total amount of flow that reaches the sink, the *waste* $W(t)$ in round t is D minus the throughput, and the *total waste* as the sum of the waste over all rounds.

Our aim is to find optimal protocols, for each type of feedback, that output a feasible solution in a minimal number of rounds or cause a minimal amount of total waste. We introduce functions that describe how well such optimal protocols perform: Let $\text{ROUNDS-BF}(n,D,C)$ denote the minimal number of rounds for binary feedback needed by any protocol to find a feasible solution for n channels, a demand D, and a network capacity C. Similarly, we define $\text{ROUNDS-TF}(n,D,C)$ for throughput feedback. The functions $\text{WASTE-BF}(n,D,C)$ and $\text{WASTE-TF}(n,D,C)$ tell us how much total waste any protocol must cause before it finds a solution. Finally, we also introduce four sibling functions that miss the third parameter C as in $\text{ROUNDS-BF}(n,D)$. For these functions the total capacity C is not known to the protocol and the protocol must find a solution or infer that $C < D$.

Motivation. The original motivation for this problem was a scenario in which the channels correspond to Internet service providers, and the goal is to distribute a given amount of traffic among the providers, without advance knowledge of the maximum rate at which each provider is willing and able to transmit our traffic [1]. On a smaller scale, a user of a peer-to-peer network who tries to service parallel download requests from different peers also faces the problem of distributing her fixed, often small, bandwidth among the peers.

In addition to the network flow interpretation, there is an interpretation of our problem as a product distribution problem. For example, suppose a publisher can produce up to D copies of a newspaper each day, and must distribute the copies to n cities, where the demand c_i at each city i is initially unknown. Our problem models the process of efficient probing for the distribution that maximizes sales.

Our Contribution. Our main results can be summarized by the inequalities shown below. In all cases, the lower bounds are established through adversary strategies. The upper bounds are established by analyzing the performance of concrete protocols. In the equations, and in the following, $o(1)$ always refers to the parameter n.

Function	Lower Bound	Upper Bound
$\text{ROUNDS-TF}(n,D)$	$\ge \big(1-o(1)\big)\frac{\ln n}{\ln\ln n}$	$\le \big(1+o(1)\big)\frac{\ln n}{\ln\ln n}$
$\text{WASTE-TF}(n,D)$	$\ge \big(1-o(1)\big)D\frac{\ln n}{\ln\ln n}$	$\le \big(1+o(1)\big)D\frac{\ln n}{\ln\ln n}$
$\text{WASTE-BF}(n,D,C)$	$\ge \big(1-o(1)\big)D\frac{\ln n}{\ln\ln n}$	$\le \big(2+o(1)\big)D\frac{\ln n}{\ln\ln n}$
$\text{ROUNDS-BF}(n,D,C)$	$\ge \big(1-o(1)\big)\big(\log_2 \frac{C}{C-D+1} - \log_2 n\big)$	$\le \log_2 \frac{C}{C-D+1} + \frac{3}{2}\log_2 n$
$\text{ROUNDS-BF}(2,D,D)$	$= \lceil \log_3 D \rceil$	$= \lceil \log_3 D \rceil$

Organization of this Paper. In Section 2 we introduce notions that are common to the analysis of all variants of the problem. Section 3 treats throughput feedback, and we establish matching upper and lower bounds, simultaneously for the minimal number of rounds and the minimal waste. In Section 4 we study binary feedback. There, we treat waste and rounds separately and give special attention to the case of two channels. We conclude with a list of open problems.

2 Basic Protocol Analysis Tools

In this section we introduce basic ideas and terminology that will be used in all of our analyses.

2.1 Maintaining the Pinning Box

For any protocol, at any point during a run of the protocol we will have gathered certain information about the (unknown) capacities of the channels. For each channel i, from the answers to the previous t queries we will have deduced an *upper bound* $h_i(t)$ and a *lower bound* $l_i(t)$ for the channel capacity c_i. Thus $c_i \in [l_i(t), h_i(t)]$, called the *pinning interval*. The cross product of the pinning intervals will be called the *pinning box*. The sum $\sum_{i=1}^{n} \big(h_i(t) - l_i(t)\big)$ will be called the *size* of the pinning box. If, at any point, $l_i(t) = h_i(t)$, we obviously know c_i. At the beginning of a run, we know the trivial bounds $l_i(0) = 0$ and $h_i(0) = \infty$. A better upper bound is given by $h_i(0) = C$, but we may not know C. The sum of the $l_i(t)$ at time t will be denoted $L(t)$. Similarly, the sum of the $h_i(t)$ will be denoted $H(t)$.

We query a vector $q(t) = (q_1, \ldots, q_n)$ at time step t. The feedback $f(t) = (f_1, \ldots, f_n)$ may allow us to improve some or perhaps even all of our pinning intervals. For binary feedback, we can perform the following updating: if $f_i = \textit{failure}$, set $h_i(t) = \min\{h_i(t-1), \lceil q_i \rceil - 1\}$; if $f_i = \textit{success}$, set $l_i(t) = \max\{l_i(t-1), \lceil q_i \rceil\}$. It may seem strange that we allow for the possibility of trying to transmit less on a channel than the lower bound or more than the upper bound. Indeed, transmitting more than the upper bound makes little sense, but transmitting less than the lower bound can be useful: The demand "saved" by not transmitting it on a certain channel might be used to probe the capacity of other channels more quickly. For throughput feedback, we can always set $l_i(t) = \max\{l_i(t-1), f_i\}$; and if $f_i < q_i$, we even know $l_i(t) = h_i(t) = f_i = c_i$.

2.2 Effects of Increasing the Capacity

In certain situations an increase in the total capacity C affects the number of rounds or the waste needed to find a solution. Intuitively, a bigger capacity C should make it easier to find a solution—or at least not harder. However, a protocol that works fine for a capacity of, say, $C = D$ might try to exploit this fact to its advantage. For example for $n = 2$ and $C = D$, if we know $l_1 = \frac{3}{4}D$, then we can conclude that the capacity on the second channel can be at most $D/4$, *but we cannot conclude this if we only know $C \geq D$.* Nevertheless, the following theorem shows that our first intuition is correct.

Theorem 2.1. *Let $D \leq C \leq C'$. Then*

$$\textsc{rounds-bf}(n, D, C) \geq \textsc{rounds-bf}(n, D, C'),$$
$$\textsc{waste-bf}(n, D, C) \geq \textsc{waste-bf}(n, D, C').$$

Proof. Let P be a protocol that minimizes the number of rounds for n channels, a demand D, and a guaranteed capacity of C. We give a protocol P' that will need at most as many rounds as P and will work for any capacity $C' \geq C$. It does not even need to know C'.

Protocol P'.

in *round* $t \leftarrow 1, 2, 3, \ldots$ ***do***
    let $q(t)$ *be the query protocol P would pose in round t*
        if it had seen the same results to our previous queries as we have seen
    query $q(t)$
    let $B := \big\{(c_1, \ldots, c_n) \mid l_i \leq c_i \leq h_i,\ \sum_{i=1}^{n} c_i = C\big\}$
    let $(m_1, \ldots, m_n) := (\min_{c \in B} c_1, \ldots, \min_{c \in B} c_n)$

if $\sum_{i=1}^{n} m_i \geq D$ ***then***

  output *some* $(d_1, \ldots, d_n)$ *with* $\sum_{i=1}^{n} d_i = D$ *and* $l_i \leq d_i \leq m_i$; ***stop***

In essence, for an unknown capacity vector $c' = (c'_1, \ldots, c'_n)$ summing up to C', Protocol P' runs Protocol P, "pretending" that the capacity is C. It interrupts the simulation once it has found a vector $m = (m_1, \ldots, m_n)$ that sums up to at least D and that, in a certain sense, lies "beneath" all vectors summing up to C inside the pinning box.

Our first claim is that the output of the protocol is, indeed, a solution. There exists a vector $c \in B$ that is componentwise below the real capacity vector c'. It can be obtained, for example, by successively dropping the components of c' to their established lower bounds until we can drop some components exactly as much as is needed to make the resulting vector c sum up to C. Then $c \in B$. Since $c \in B$, the vector m will be componentwise below c. Since the output is componentwise below m in turn, we conclude that the output is componentwise below the capacity vector c' and is hence a solution.

Our second claim is that Protocol P' runs for at most $T := \textsc{rounds-bf}(n,D,C)$ rounds. Consider the situation the protocol faces at round T. The crucial observation at this point is that all elements of the set B produce the exact same answers to all the queries the Protocol P' (and hence also P) has posed until now. Since Protocol P always finishes within T rounds, it must be able to output a solution that satisfies all elements of B. But such a solution must necessarily lie beneath m, which must thus sum up to D. Thus $\sum_{i=1}^{n} m_i \geq D$.

For the claim $\textsc{waste-bf}(n,D,C) \leq \textsc{waste-bf}(n,D,C')$, just note that Protocol P' also wastes at most as much as Protocol P does. □

Corollary 2.2.

$$\textsc{rounds-bf}(n,D,D) = \textsc{rounds-bf}(n,D),$$
$$\textsc{waste-bf}(n,D,D) = \textsc{waste-bf}(n,D).$$

The inequality $\textsc{rounds-bf}(n,D,C) \leq \textsc{rounds-bf}(n,D,C')$ is a proper inequality in many cases. For example, we will see that $\textsc{rounds-bf}(n,D,D) = \Theta(\log D)$ but $\textsc{rounds-bf}(n,D,2D) = \Theta(1)$ for fixed n.

For throughput feedback, the situation is simpler.

Theorem 2.3. *For all $C \geq D$ we have*

$$\textsc{rounds-tf}(n,D,C) = \textsc{rounds-tf}(n,D),$$
$$\textsc{waste-tf}(n,D,C) = \textsc{waste-tf}(n,D).$$

Proof. As in the previous proof, we take a protocol P that minimizes the number of rounds or the waste and construct a protocol P' that simulates this protocol. Only this time, we stop the simulation when the sum of the lower bounds exceeds D. Observe that until this happens, there always exists a capacity vector summing up to D that produces the exact same feedback as the one we saw. Furthermore, for this capacity vector, at least one channel will not yet have been upper-bounded. Thus we can also find a vector summing up to any value $C > D$ that produces the exact same feedback as the one we saw. □

3 Throughput Feedback

In this section we establish matching upper and lower bounds on the number of rounds and the waste needed to find a solution when we get throughput feedback. The upper bound is established by explicitly describing and analyzing a concrete protocol, the lower bound is shown using an adversary argument. The results of this section can be summed up by the following equations:

$$\big(1-o(1)\big)\frac{\ln n}{\ln\ln n} \le \textsc{rounds-tf}(n,D) \le \big(1+o(1)\big)\frac{\ln n}{\ln\ln n},$$
$$\big(1-o(1)\big)D\frac{\ln n}{\ln\ln n} \le \textsc{waste-tf}(n,D) \le \big(1+o(1)\big)D\frac{\ln n}{\ln\ln n}.$$

3.1 Upper Bounds on Rounds and Waste

For throughput feedback, the capacity of a channel is determined as soon as a query exceeds it ("*overshoots*"). We propose the following protocol that seeks to overshoot as many channels as possible as quickly as possible.

Protocol 3.1 (Upper Bound Protocol).

$g_0 \leftarrow n$
in *round* $t \leftarrow 1,2,3,\dots$ ***do***
    $q_1 = \cdots = q_{g_{t-1}} := D/g_{t-1}$
    query $q(t) = (q_1,\dots,q_{g_{t-1}},0,\dots,0)$
    $g_t \leftarrow$ *the number of channels with capacity at least* D/g_{t-1}
    reorder *the channels such that exactly the first* g_t *channels*
        have capacity at least D/g_{t-1}
    if $g_t = 0$ *or* $H(t) = D$ *or* $L(t) = D$ ***then stop***

Observe that the protocol, during its course, only uses channels whose capacity has not been determined yet, and distributes the flow evenly among them. It stops when it has determined all channel capacities and also when it has found a flow distribution that is compatible with them. We next give a bound on the number of rounds the protocol needs to finish.

Theorem 3.2. $\textsc{rounds-tf}(n,D) \le \big(1+o(1)\big)\frac{\ln n}{\ln\ln n}$.

Proof. We show that Protocol 3.1 will find solutions in no more than $\big(1+o(1)\big)\frac{\ln n}{\ln\ln n}$ rounds. The main idea of the proof is that the longer it takes to overshoot the channels, the quicker the known capacity increases. Consider a run of the protocol and let T be the number of queries issued. In the following, the index t will be understood to range from 1 to $T-1$, unless stated otherwise.

Let $\alpha_t := g_t/g_{t-1}$. After the first round, g_1 channels are known to have capacity at least D/g_0, so $L(1) = \alpha_1 D$. Similarly, after round $t \ge 2$, the known capacity on the g_t channels that are not overshot increases from D/g_{t-2} to D/g_{t-1}. Thus,

$$L(t) - L(t-1) \ge g_t(D/g_{t-1} - D/g_{t-2}) = \alpha_t(1-\alpha_{t-1})D. \tag{1}$$

Let $\max \in \{1,\dots,T-1\}$ be an index such that $\alpha_{\max} \ge \alpha_t$ for all t. Telescoping (1), we get

$$L(T-1)/D \ge \alpha_1 + \sum_{t\ge 2}\alpha_t(1-\alpha_{t-1}) \tag{2}$$
$$= \alpha_{\max} + \sum_{t=1}^{\max-1}\alpha_t(1-\alpha_{t+1}) + \sum_{t=\max+1}^{T-1}\alpha_t(1-\alpha_{t-1}) \tag{3}$$
$$\ge \alpha_{\max} + (1-\alpha_{\max})\sum_{t\ne\max}\alpha_t. \tag{4}$$

Since the protocol does not stop after query $T-1$, we know $L(T-1) < D$. Thus, by (3), $\alpha_{\max} \leq L(T-1)/D < 1$. We may therefore divide by $1-\alpha_{\max}$ and from (4) we get

$$\sum_{t \neq \max} \alpha_t < 1. \tag{5}$$

Another consequence of the fact that the protocol does not stop after query $T-1$ is that $g_{T-1} \geq 1$, where $g_{T-1} = n\prod_{t=1}^{T-1} \alpha_t$. Thus,

$$\prod_{t \neq \max} \alpha_t > \prod_{t=1}^{T-1} \alpha_t \geq 1/n. \tag{6}$$

Applying the inequality of the arithmetic and geometric means, from (5) and (6) we get

$$1/n < \prod_{t \neq \max} \alpha_t \leq \left(\frac{\sum_{t \neq \max} \alpha_t}{T-2} \right)^{T-2} < (T-2)^{-(T-2)}, \tag{7}$$

and consequently, $(T-2)^{T-2} \leq n$. Therefore $T \leq \big(1+o(1)\big)\frac{\ln n}{\ln\ln n}$ and the claim follows. □

Our upper bound on the number of rounds has the following immediate corollary, which is proved by observing that in any round one can waste at most D.

Corollary 3.3. $\text{WASTE-TF}(n,D) \leq \big(1+o(1)\big)D\frac{\ln n}{\ln\ln n}$.

3.2 Lower Bounds on Rounds and Waste

We give lower bounds that match the upper bounds above, first on the waste.

Theorem 3.4. $\text{WASTE-TF}(n,D) \geq \big(1-o(1)\big)D\frac{\ln n}{\ln\ln n}$.

Proof. Consider a optimal protocol. We describe an adversary strategy that causes it to waste at least the amount stated in the theorem.

The Capacity Distribution. Let $\alpha := 1/\ln n$ and $T := \left\lceil \frac{\ln n}{\ln\ln n} \right\rceil$.

After each query, as long as there is at least one channel left, the adversary fixes the capacity of all but an α-fraction of the channels. More precisely, after the t first queries the capacity of $g_t := \lceil \alpha^t n \rceil$ channels has not yet been fixed. Observe that $\alpha^{T-1}n > 1$, but $\alpha^T n \leq 1$, so the number of queries handled that way is $T-1$. After T queries, only one channel is left and its capacity is chosen as the difference between C and the total capacity already assigned. The feedback is always given consistently with all capacity assignments that extend the fixed capacities.

If the capacity of two channels is fixed after the same query, the capacities shall be the same. Thus, there are T different capacity values to be chosen. We denote them, abusing notation, by $c_1, \ldots, c_T$, where c_i is the capacity of the channels fixed after query i. Specifically, let $c_1 := 0$ and $c_t := \frac{D}{(1-\alpha)\alpha^{t-2}n}$ for $2 \leq t \leq T-1$.

For convenience, we denote the number of channels fixed after query t by

$$r_t := g_{t-1} - g_t \leq \alpha^{t-1}n + 1 - \alpha^t n \leq \left(1 + \frac{1}{\alpha^{t-1}n}\right)\alpha^{t-1}(1-\alpha)n,$$

whence the newly assigned capacity after query t is

$$r_t \cdot c_t \leq \left(1 + \frac{1}{\alpha^{t-1}n}\right)\alpha D \leq 2\alpha D. \tag{8}$$

An immediate consequence of (8) is

$$L(T-1) = \sum_{t=1}^{T-1} r_t c_t \leq 2\alpha(T-1)D = o(D),$$

so that the protocol has not located enough capacity after any of the first $T-1$ queries. Moreover, there is always at least one unbounded channel left. Therefore the protocol continues up to the Tth query. Note that before that query there is only one unbounded channel left, and that that channel is given the remaining capacity value $C - \sum_{t=1}^{T} r_t c_t \geq \big(1-o(1)\big)D$.

Assignment of the Capacities. Since the capacity distribution is not dependent on the protocol, it is convenient to describe the information state of the protocol before query t as an assignment of the capacities $c_1, \ldots, c_{t-1}$ to a selection of $n - g_{t-1} = r_1 + \cdots + r_{t-1}$ channels. As an invariant we shall maintain that none of the channels whose capacity has not yet been fixed has been queried with an amount of c_t or more.

The invariant holds before the first query. Given the tth query $q(t) = (q_1, \ldots, q_n)$, using the invariant we may assume without loss of generality that $q_1 \leq \cdots \leq q_{g_{t-1}}$. The adversary assigns the capacity c_t to channels $g_t+1, g_t+2, \ldots, g_{t-1}$. We need to verify that the invariant still holds. Since $D \geq q_{g_t} + \cdots + q_n \geq (r_t+1)q_{g_t}$, where

$$r_t + 1 = g_{t-1} - g_t + 1 > \alpha^{t-1}n - \alpha^t n = (1-\alpha)\alpha^{t-1}n,$$

we have $g_{t'} \leq g_t < \frac{D}{(1-\alpha)\alpha^{t-1}n} = c_{t+1}$ for $1 \leq t' \leq g_t$.

The information state of the protocol before query t is *not more* than stated in the invariant. Although the protocol might gather less information in the first t queries, we can just assume that the adversary gives away the difference in information.

The Total Waste. Let $P(t)$ denote the throughput in query t. By the definition of throughput,

$$P(t) \leq \sum_{j=1}^{g_t} q_j + r_t c_t + \cdots + r_1 c_1 \leq g_t / g_{t-1} \cdot D + 2(t-1)\alpha D.$$

The second inequality follows from the order on the q_j and inequality (8).

Since $g_t/g_{t-1} \leq (1 + \frac{1}{\alpha^t n})\alpha \leq 2\alpha$, we get $P(t) \leq 2t\alpha D$, and thus a total waste of

$$\sum_{t=1}^{T-1} (D - P^t) \geq (T-1)D - 2\binom{T}{2}\alpha D = TD\big(1 - T\alpha\big),$$

which proves the theorem. □

Once more, by observing that in any round we can waste at most D, we get a corollary.

Corollary 3.5. ROUNDS-TF$(n,D) \geq \big(1-o(1)\big)\frac{\ln n}{\ln\ln n}$.

4 Binary Feedback

In this section we study upper and lower bounds for flow distribution when we get binary feedback. We first study the waste function WASTE-BF(n,D,C) and show that it behaves similarly to the corresponding waste function for throughput feedback. Next, we review results from the literature on the function ROUNDS-BF(n,D,D) and then extend these results to ROUNDS-BF(n,D,C). Unlike ROUNDS-TF(n,D,C), for binary feedback the number of rounds does not only depend on n and D, but also on C. While the upper and lower bounds that we establish for ROUNDS-BF(n,D,C) do not quite match for general n, we solve the special case of two channels completely: interestingly, ROUNDS-BF$(2,D,D) = \lceil \log_3 D \rceil$.

4.1 Minimizing Waste

Similarly to the treatment of throughput feedback, we establish upper and lower bounds for the waste needed to find a solution. We show the following:

$$\big(1-o(1)\big)D\frac{\ln n}{\ln\ln n} \le \text{WASTE-BF}(n,D,C) \le \big(2+o(1)\big)D\frac{\ln n}{\ln\ln n}.$$

The first bound follows from Theorem 3.4. To prove the second bound, we proceed as follows: We use Protocol 3.1 once more. While it is an optimal waste minimization protocol for throughput feedback, for binary feedback it typically leaves us with some pinning box. We use a second protocol to reduce the pinning box's size to zero. Intriguingly, this second protocol is the *proportional allocation protocol*, which was originally introduced in [1] in the context of *round* minimization. We show that this protocol reduces the size of the pinning box to zero without wasting more than the total size of the pinning intervals. Finally, we show that when we start the proportional allocation protocol, the pinning box has size at most $D\ln n/\ln\ln n$.

Protocol 4.1 (Proportional Allocation, [1]).

foreach $i \in \{1,\dots,n\}$
    $h_i \leftarrow \min\{h_i, D\}$
in *round* t ***do***
    $\rho \leftarrow \frac{D-L(t-1)}{H(t-1)-L(t-1)}$
    foreach $i \in \{1,\dots,n\}$ ***do***
        $q_i(t) \leftarrow l_i(t-1) + \rho\big(h_i(t-1) - l_i(t-1)\big)$
    query $\big(q_1(t),\dots,q_n(t)\big)$
    if $H(t) = D$ *or* $L(t) = D$ ***then output*** *last query;* ***stop***

Theorem 4.2. *Let Protocol 4.1 be started at step t_0 with a certain pinning box already established. Then it will find a solution wasting no more than $H(t_0) - L(t_0)$.*

Proof. Let $\Delta(t) := H(t) - L(t)$ denote the total pinning interval gap. Our aim is to show the following claim: *If the protocol wastes $W(t)$ in round t, we have $\Delta(t+1) \le \Delta(t) - W(t)$.* In other words, we reduce the pinning interval gap by at least the amount we waste. If this claim holds, we clearly cannot waste more than $\Delta(t_0)$ before the gap drops to zero.

In each round t we distinguish two cases, depending on whether $\rho \le 1/2$ or $\rho > 1/2$ in this round. For the case $\rho \le 1/2$, consider the values $w_i := \max\{0, q_i - c_i\}$. We claim $W(t) = \sum_{i=1}^n w_i$. This can be seen as follows:

$$W(t) = D - \sum_{i=1}^n \min\{c_i, q_i\} = D - \sum_{i=1}^n \big(q_i - \max\{0, q_i - c_i\}\big) = \sum_{i=1}^n w_i.$$

We used the fact that Protocol 4.1 always distributes the complete demand D, that is, $\sum_{i=1}^n q_i = D$. Consider a channel i with $w_i > 0$ and thus $q_i > c_i$. Since the query was a failure, the upper bound h_i will be decreased to q_i. Thus the pinning interval changes from $[l_i, h_i]$ to $[l_i, q_i]$ and its size changes from $h_i - l_i$ to $q_i - l_i = \rho(h_i - l_i)$. Thus, channel i causes the gap Δ to shrink by at least $(1-\rho)(h_i - l_i)$ while it causes a waste of at most $w_i = q_i - c_i \le q_i - l_i = \rho(h_i - l_i)$. Since this argument is true for all channels, we conclude that $W(t)$ is less than the decrease of the pinning interval gap.

For the case $\rho > 1/2$, we argue similarly, but we now consider the values $w_i' := \max\{0, c_i - q_i\}$. The sum of these values is an upper bound on the waste:

$$W(t) = D - \sum_{i=1}^n \min\{c_i, q_i\} = D - \sum_{i=1}^n \big(c_i - \max\{0, c_i - q_i\}\big) = D - C + \sum_{i=1}^n w_i' \le \sum_{i=1}^n w_i'.$$

It remains to argue that on each channel we decrease the size of the pinning interval by at least w'_i. Suppose $w'_i > 0$. Then $c_i > q_i$ and the pinning interval changes from $[l_i, h_i]$ to $[q_i, h_i]$. Its size changes from $h_i - l_i$ to $h_i - q_i = (1-\rho)(h_i - l_i)$ and thus its size reduces by at least $\rho(h_i - l_i)$. Since $w'_i = c_i - q_i \le h_i - q_i = (1-\rho)(h_i - l_i)$ and since $\rho > 1/2$, we conclude that the reduction of the interval size is larger than the waste. □

Theorem 4.3. $\text{WASTE-BF}(n,D,C) \le \big(2+o(1)\big)D\frac{\ln n}{\ln\ln n}$.

Proof. We run Protocol 3.1, followed by Protocol 4.1. By Corollary 3.3, the waste at the end of the first protocol is $\big(1+o(1)\big)D\ln n/\ln\ln n$. Observe that an upper bound is established when a channel is being overshot and that Protocol 3.1 overshoots each channel exactly once. Consider the channels that are overshot in round t. For each of them (there are $g_{t-1} - g_t$ many), the protocol queries D/g_{t-1} and gets that value as an upper bound on their capacity. These upper bounds cumulate to $D(1 - g_t/g_{t-1}) \le D$. Given the bound on the number of queries from Theorem 3.2, we conclude $H(t_0) - L(t_0) \le H(t_0) \le \big(1+o(1)\big)D\ln n/\ln\ln n$. Theorem 4.2 now yields the claim. □

4.2 Minimizing Rounds

In the previous section we saw that for the total waste it makes only little difference whether binary feedback and throughput feedback is used: using binary feedback at most doubles the waste. In this section we see that the situation is quite different for rounds. For throughput feedback the optimal number of rounds is $\ln n/\ln\ln n$. In particular, this number does not depend on either D or C. As we show in the following, for binary feedback the optimal number of rounds depends both on D and on C. In particular, an increased total capacity C allows us to find solutions more quickly. In detail, we show the following:

$$\big(1-o(1)\big)\left(\log_2\frac{C}{C-D+1} - \log_2 n\right) \le \text{ROUNDS-BF}(n,D,C) \le \log_2\frac{C}{C-D+1} + \frac{3}{2}\log_2 n.$$

As we mentioned earlier, Chandrayana et al. have proposed the proportional allocation protocol, Protocol 4.1, as a protocol for minimizing the number of rounds. They prove the following fact:

Fact 4.4 ([1]). $\text{ROUNDS-BF}(n,D,D) \le \log_2 D + \frac{\log_2 n}{2}$.

Chandrayana et al. prove a lower bound that matches this upper bound in the sense that they show $\text{ROUNDS-BF}(n,D,D) = \theta(\log_2 D)$ for fixed n. However, the protocol is far from optimal if C is larger than D and if we know this. Suppose for example, that $n = 2$ and we know $C = 2D$. In this situation, if $c_1 = C$ and $c_2 = 0$, Protocol 4.1 will need $\log_2 D$ rounds to find the solution $(D,0)$. There is a much faster protocol: In the first round, query $(D/2, D/2)$. If this query is a solution, we are done. Otherwise, on at least one channel our query will succeed, say on the second one. But then $(0,D)$ is a solution. Thus $\text{ROUNDS-BF}(2,D,2D) = 2$ and this is true *for all* D.

We present a modification of Protocol 4.1 that finds a solution in a number of rounds that depends on n but not on D, if we know $C = \alpha D$ for some constant $\alpha > 1$. The protocol will need at most $\log_2\frac{\alpha}{\alpha-1} + \frac{3}{2}\log_2 n$ rounds in this situation.

Protocol 4.5 (Scaled Proportional Allocation). *Let* $\Delta := \lfloor (C-D)/n \rfloor + 1$ *and let* $D' := \lceil D/\Delta \rceil$*. Scaled proportional allocation simulates Protocol 4.1 with the following modifications:*

1. *The demand we try to distribute is* D' *instead of* D.
2. *Whenever proportional allocation wishes to make a query* $(q'_1, \ldots, q'_n)$*, query* $(\Delta q'_1, \ldots, \Delta q'_n)$ *instead.*

Theorem 4.6. ROUNDS-BF$(n,D,C) \le \log_2 \frac{C}{C-D+1} + \frac{3}{2}\log_2 n$.

Proof. Consider the actual capacities $(c_1,\dots,c_n)$ and let $c'_i := \lfloor c_i/\Delta \rfloor$. A run of Protocol 4.5 will produce the same queries as running the original Protocol 4.1 for the capacity vector $(c'_1,\dots,c'_n)$ and for the demand D': We have $\Delta q'_i \le c_i$ if and only if $\Delta q'_i \le \Delta \lfloor c_i/\Delta \rfloor$, which is in turn equivalent to $q'_i \le \lfloor c_i/\Delta \rfloor$. Thus a query $\Delta q'_i$ will be answered with *success* in Protocol 4.5 if and only if the query q'_i is answered the same way in Protocol 4.1 for the scaled capacities and the scaled demand.

There exists a solution with respect to the capacities $(c'_1,\dots,c'_n)$ and the demand D' since

$$\sum_{i=1}^{n} c'_i = \sum_{i=1}^{n} \left\lfloor \frac{c_i}{\Delta} \right\rfloor \ge \sum_{i=1}^{n} \frac{c_i - \Delta + 1}{\Delta} = \frac{C - n\Delta + n}{\Delta} = \frac{D}{\Delta}.$$

The left-hand side is an integer and we even have $\sum_{i=1}^{n} c'_i \ge \lceil D/\Delta \rceil = D'$.

So far, we have shown that Protocol 4.5 will need as much time to find a solution as Protocol 4.1 will need for the scaled capacity vector and the scaled demand. By Fact 4.4, we will find a solution in time

$$\begin{aligned}\log_2 D' + \frac{\log_2 n}{2} &= \log_2 \left\lceil \frac{D}{\lfloor (C-D)/n \rfloor + 1} \right\rceil + \frac{\log_2 n}{2} = \log_2 \left\lceil \frac{nD}{n\lfloor (C-D)/n + 1 \rfloor} \right\rceil + \frac{\log_2 n}{2} \\ &\le \log_2 \left\lceil \frac{nD}{C-D+1} \right\rceil + \frac{\log_2 n}{2} \le \log_2 \frac{C}{C-D+1} + \frac{3}{2}\log_2 n.\end{aligned}$$

□

Theorem 4.7. ROUNDS-BF$(n,D,C) \ge \big(1 - o(1)\big)\big(\log_2 \frac{C}{C-D+1} - \log_2 n\big)$.

Proof. We present an adversary strategy against an optimal protocol for given numbers n, D, and C. The adversary keeps track of a set X of capacity vectors summing up to C that are consistent with all the answers the adversary has provided until now. Initially, X contains all vectors of nonnegative integers summing up to C and thus has size $\binom{C+n-1}{n-1}$. When the protocol poses a query, the 2^n possible answers vectors partition X into 2^n sets whose elements are consistent with one answer vector. At least one of these sets has size at least $|X|/2^n$ and the adversary returns the answer vector corresponding to this set.

We claim that the protocol cannot produce its final output before $|X|$ has dropped to $\binom{C-D+n-1}{n-1}$. To see this, note that every vector summing up to D is componentwise below at most $\binom{C-D+n-1}{n-1}$ many vectors in X. We conclude that the number T of rounds needed by the optimal protocol to produce its solution must satisfy $\binom{C+n-1}{n-1} \,/\, (2^n)^T \le \binom{C-D+n-1}{n-1}$ and thus

$$T \ge \frac{1}{n}\log_2 \frac{\binom{C+n-1}{n-1}}{\binom{C-D+n-1}{n-1}} \ge \frac{n-1}{n}\log_2 \frac{C+n-1}{C-D+n-1} \ge \frac{n-1}{n}\log_2 \frac{C}{n(C-D+1)}.$$

The last term equals $\big(1-o(1)\big)\big(\log_2 \frac{C}{C-D+1} - \log_2 n\big)$, which proves the claim. □

4.3 Minimizing Rounds for Two Channels

The upper and lower bounds proved in the previous section do not quite match: the upper bounds contain a positive $\log_2 n$ term, the lower bounds a negative $\log_2 n$ term. As a first step toward closing this gap, we completely solve the problem for $n = 2$: the number of rounds needed to find a solution using binary feedback is $\log_3 D$ rounds for $C = D$. This result is a bit surprising since even a binary search needs $\log_2 D$ rounds to find a distribution. The protocol that achieves the bound of $\log_3 D$ is the following:

Protocol 4.8 (Two Channel Protocol).

$l \leftarrow 0$, $h \leftarrow D$
in *round* t ***do***
    $q_1 \leftarrow (h-l)/3+l$
    $q_2 \leftarrow (h-l)/3+D-h$
    query (q_1,q_2) ***receiving*** (f_1, f_2)
    if $H(t) = D$ *or* $L(t) = D$ ***then output*** *last query;* ***stop***
    if $f_1 = failure$ ***then*** $h \leftarrow q_1$ ***else*** $l \leftarrow q_1$
    if $f_2 = failure$ ***then*** $l \leftarrow \max\{l, D-q_2\}$ ***else*** $h \leftarrow \min\{h, D-q_2\}$

Theorem 4.9. $\text{ROUNDS-BF}(2,D) = \log_3 D$.

Proof. By Corollary 2.2 it suffices to show $\text{ROUNDS-BF}(2,D,D) = \log_3 D$. We begin with an adversary strategy that ensures that any fixed protocol cannot find the solution in less than $\log_3 D$ rounds. This will show the inequality $\text{ROUNDS-BF}(2,D,D) \geq \log_3 D$.

The aim of the adversary is to keep the protocol in the dark about the capacity c_1. The adversary keeps track of a pinning interval $[l,h]$ that gets smaller in each round. In any round t, the answers of the adversary up to then will be consistent with every capacity vector (c_1,c_2) with $c_1 \in [l,h]$ and $c_2 = D-c_1$. Initially, $l = 0$ and $h = D$, which clearly fulfills the requirements. In round t, consider a query $q(t) = (q_1,q_2)$ and consider where the two numbers q_1 and $D-q_2$ lie in the interval $[l,h]$. They can split the interval into at most three intervals and at least one of them must have size at least $(h-l)/3$. The adversary answers such that any value within this largest interval is permissible. For simplicity, assume $l \leq q_1 \leq D-q_2 \leq h$—other cases are similar. Then, in detail, if the largest interval is $[l,q_1]$, the adversary answers $(\mathit{failure}, \mathit{success})$. If the largest interval is $[q_1, D-q_2]$, the adversary answers $(\mathit{success}, \mathit{success})$. If the largest interval is $[D-q_2, h]$, the adversary answers $(\mathit{success}, \mathit{failure})$. In each round, the size of the interval is reduced by a factor of at most 3. Thus, only after $\log_3 D$ rounds the adversary will finally have to settle on a capacity distribution.

To prove the inequality $\text{ROUNDS-BF}(2,D,D) \leq \log_3 D$, consider Protocol 4.8. It implements a strategy against the just-given adversary by keeping track of the interval $[l,h]$ for which it knows $c_1 \in [l,h]$ and thus $c_2 \in [D-h, D-l]$. In each round it poses two queries (q_1,q_2) such that q_1 and $D-q_2$ cut the interval into three equal parts. For every possible answer vector (f_1, f_2) the interval size will be reduced by a factor of 3, which proves the claim. □

5 Conclusion and Open Problems

In this paper we proposed a framework for studying different ways of distributing a flow in a simple network with unknown capacities. We studied two kinds of feedback, namely binary and throughput feedback. For the latter type of feedback we presented a protocol that is optimal both with respect to the number of rounds and the waste produced. For binary feedback there is still a gap between the upper and lower bounds when the number n of channels is also taken into account. For the special case of two channels we gave an optimal protocol that outperforms binary search.

Experimental work done by Chandrayana et al. [1] has shown that the (unscaled) protocol allocation protocol performs well on real data with respect to the number of rounds needed. Our theoretical work backs these findings, but we showed that scaling can significantly improve the performance of the protocol if the available capacity is larger than the demand. The proportional allocation protocol is also part of our near-optimal protocol for waste minimization. This opens the intriguing possibility that the scaled proportional allocation protocol might be an optimal protocol for minimizing both rounds and waste.

We did not address the computational complexity of protocols in the present paper; protocols conceptually had arbitrarily much computational power. However, reviewing the protocols for which we showed optimality, we see that they are both easy to implement and have low computational complexity.

We mention the following open problems, which we suggest for further research:

1. In the binary feedback model, find an algorithm that is optimal, up to a factor $1+o(1)$, with respect to both total waste and number of rounds.

2. In the binary feedback model, improve the upper and lower bounds on the optimal number of rounds when a lower bound is given on the total capacity C.

3. Study dynamic versions of our problems, in which the capacities may change from round to round. This can be done either under deterministic constraints on the changes of capacity from round to round, or under a probabilistic model of the fluctuation of capacities. For the case $n = 1$, some results in the framework of competitive analysis of on-line algorithms are given in [2].

4. Determine the optimal waste for a model in which the throughput on channel i is q_i if $q_i \leq c_i$, and 0 if $q_i > c_i$. This model is suggested by certain Internet congestion control protocols in which, whenever a packet is dropped in a round, it cannot be guaranteed that any packets are delivered. For the case $n = 1$, this model is studied in [2].

5. Our problem suggests a generalization of the classic Twenty Questions scenario in which a player attempts to efficiently identify an unknown object by asking yes / no questions. In the generalization, the player's goal is to identify several objects concurrently by asking a question about each object in each round, but each question has a cost, and a limit is placed on the total cost that may be expended in a round. Explore further examples of this generalized Twenty Questions framework.

References

[1] Kartikeya Chandrayana, Yin Zhang, Matthew Roughan, Shubho Sen, and Richard Karp. Search game in inter-domain traffic engineering. Manuscript, 2004.

[2] R. Karp, E. Koutsoupias, C. Papadimitriou, and S. Shenker. Combinatorial optimization in congestion control. In *Proceedings of the 41th Annual Symposium on Foundations of Computer Science*, pages 66–74, Redondo Beach, CA, 2000.