

# Online Control Message Aggregation in Chain Networks<sup>\*</sup>

Marcin Bienkowski<sup>1</sup>, Jaroslaw Byrka<sup>1</sup>, Marek Chrobak<sup>2</sup>, Łukasz Jeż<sup>1,3</sup>, Jiří Sgall<sup>4</sup>, and Grzegorz Stachowiak<sup>1</sup>

<sup>1</sup> Institute of Computer Science, University of Wrocław, Poland.

<sup>2</sup> Department of Computer Science, University of California at Riverside, USA.

<sup>3</sup> Dept. of Computer, Control, and Management Engineering, Sapienza University of Rome, Italy.

<sup>4</sup> Computer Science Institute, Faculty of Mathematics and Physics, Charles University, Czech Republic.

**Abstract.** In the Control Message Aggregation (CMA) problem, control packets are generated over time at the nodes of a tree  $T$  and need to be transmitted to the root of  $T$ . To optimize the overall cost, these transmissions can be delayed and different packets can be aggregated, that is a single transmission can include all packets from a subtree rooted at the root of  $T$ . The cost of this transmission is then equal to the total edge length of this subtree, independently of the number of packets that are sent. A sequence of transmissions that transmits all packets is called a *schedule*. The objective is to compute a schedule with minimum cost, where the cost of a schedule is the sum of all the transmission costs and delay costs of all packets. The problem is known to be  $\text{NP}$ -hard, even for trees of depth 2. In the online scenario, it is an open problem whether a constant-competitive algorithm exists.

We address the special case of the problem when  $T$  is a chain network. For the online case, we present a 5-competitive algorithm and give a lower bound of  $2 + \phi \approx 3.618$ , improving the previous bounds of 8 and 2, respectively. Furthermore, for the offline version, we give a polynomial-time algorithm that computes the optimum solution.

## 1 Introduction

In the Control Message Aggregation (CMA) problem, introduced in [6], we are given a tree  $T$  whose edges have positive *lengths*. Over time, packets are generated at the nodes of  $T$ . Each packet is specified by a pair  $(t, v)$ , where  $t$  is the injection time of this packet and  $v \in T$  the vertex where the packet is injected. All packets must be transmitted to the root of  $T$ , although not necessarily immediately; to reduce cost, packets can be delayed and different packets can be aggregated into a single transmission. A transmission is defined as a subtree

---

<sup>\*</sup> Research partially supported by NSF grants CCF-1217314 and OISE-1157129, MNiSW grant no. N N206 368839, 2010-2013, EU ERC project 259515 PAAI, CE-ITI (project P202/12/G061 of GA ČR), and grant IAA100190902 of GA AV ČR.

rooted at the root of  $T$  that transports *all* the packets currently contained in its nodes. The cost of this transmission is then equal to the total edge length of this subtree, independently of the number of packets that are sent. Delaying a transmission of a packet incurs cost equal to the time it waits for the transmission. A *schedule* is specified by a sequence of transmissions that transmit all packets to the root. The cost of a schedule is the sum of its transmission costs and the waiting costs of all packets. The objective is to find a schedule of minimum cost.

In reality, what we refer to as *packets* in the paper are abstractions of control messages acknowledging the receipt of some network packets from a communication stream. This motivates the assumptions of our cost model, in particular the fact that the transmission cost does not depend on the number of packets involved. The reason is that in practice acknowledgement messages are very small, so the cost of including them in the transmission is negligibly small in comparison to the overhead of sending it. More generally, the CMA problem can model transportation problems where large quantities of small items need to be shipped to a common destination, incurring both the transportation and delay costs.

While it is possible to consider the problem in the *offline* scenario, where the algorithm knows the whole input sequence in advance, the *online* model more accurately reflects the constraints that arise in practice. In the online model, packets arrive as time passes, with each packet  $(t, v)$  arriving at time  $t$ , and at each time step the algorithm needs to decide whether and what to transmit. In the online model, one can further distinguish two versions: distributed algorithms, where the nodes make local decisions independently, and centralized, *full information* algorithms [2], that have complete knowledge about the input sequence so far and the current state.

**Previous work.** Khanna et al. [6] defined the CMA problem and gave an  $O(\log \alpha)$ -competitive online algorithm, where  $\alpha$  is the sum of all edge lengths. (Their algorithm works under an additional technical assumption that each packet has to wait at least one time unit.) The CMA problem on a single-edge tree is equivalent to the prominent TCP acknowledgement problem. The online version of TCP acknowledgement has been essentially solved: the optimal competitive ratios for deterministic and randomized algorithms are, respectively, 2 and  $e/(e - 1) \approx 1.582$  [4,5].

While finding better bounds on the competitive ratio remains an open problem,  $O(1)$ -competitive online algorithms were presented for some special classes of trees. For *flat trees*, where the tree has depth two and the root has only one child, the problem becomes equivalent to the Joint Replenishment Problem (JRP) with linear penalties, well studied in Operations Research. For this version, Brito et al. [2] gave a 5-competitive algorithm. The ratio was later improved to 3 by Buchbinder et al. [3], who also presented a lower bound of 2.64. These results apply in fact to arbitrary trees of depth two, as such trees can be decomposed into flat trees that are processed independently. In the offline case, Arkin et al. [1] proved that JRP is NP-hard and a 1.8-approximation algorithm was given by Levi et al. [7,8,9].

Another previously studied case is that of chain networks, where each non-leaf node has exactly one child. This version was introduced by Brito et al. [2], who presented an 8-competitive online algorithm. No online algorithm can be better than 2-competitive in this case, due to the lower bound for the TCP-acknowledgement problem [4].

**Our contributions.** Following the work in [2], we study online and offline algorithms for Control Message Aggregation on chain networks. Three results are provided. In the online case, we focus on centralized algorithms. For this model we give a 5-competitive algorithm and we prove a lower bound of  $2 + \phi \approx 3.618$  (where  $\phi = (1 + \sqrt{5})/2$  is the golden ratio). Both results improve the previously known bounds, described above. As we show, the analysis of our online algorithm is tight. For the offline case, we provide an algorithm that computes an optimal schedule in polynomial time.

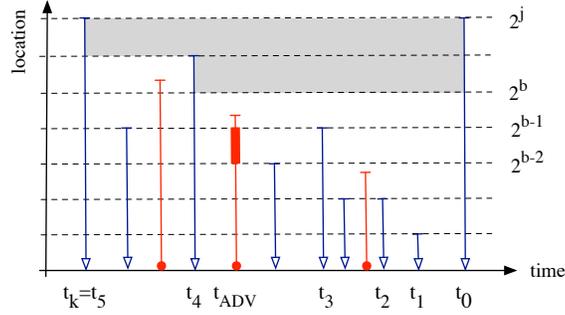
## 2 Preliminaries

Throughout the paper, we think of a chain network as a half-line  $\mathbb{R}^+$ , consisting of all non-negative real numbers. Slightly abusing the terminology, we will refer to it as *the line*. Any  $x \in \mathbb{R}^+$  represents a node of the network at distance  $x$  from the origin. Packets arrive over time and need to be transmitted to the destination at point 0. A transmission from  $x$  at time  $t$  sends all packets from the interval  $(0, x]$ , and the cost of this transmission is  $x$ . (Without loss of generality, we may assume that no packet ever arrives at point 0 as they might be transmitted immediately at zero cost.) We consider continuous time, that is packets can be injected and transmitted at arbitrary real-valued times. We also allow packets to have non-negative weights. The penalty function for waiting is assumed to be linear, that is a packet of weight  $w$  injected at time  $a$  and transmitted at time  $t$  pays the cost  $w(t - a)$  for waiting. The weights do not affect the cost of transmissions. Thus, for the purpose of computing cost,  $w$  unit-weight packets arriving at the same time are equivalent to one packet of weight  $w$ . Note that our lower bounds use only integer weights, and thus can be trivially simulated using non-weighted packets.

## 3 An Online 5-Competitive Algorithm

In this section, we present our 5-competitive algorithm for Control Message Aggregation on chain networks.

For any  $x \in \mathbb{R}^+$  and time  $t$ , denote by  $\text{twc}_t[x]$  the total waiting cost of the packets currently at  $x$ , starting at their arrival until time  $t$ . We generalize this notation to intervals: for example  $\text{twc}_t(x, y]$  is the total waiting cost in the interval  $(x, y]$ , that includes  $y$  but not  $x$ , etc. We will drop subscript  $t$  if its value is clear from the context or not relevant. The *length* of a transmission from point  $x$  is equal to  $x$ . For any time  $t$ ,  $\text{trl}(t)$  denotes the length of a transmission at time  $t$ ; we let  $\text{trl}(t) = 0$  if there is no transmission at  $t$ . For the analysis, we define an analogous notion of  $\text{trl}_{\text{ADV}}(t)$  for adversarial transmissions.



**Fig. 1.** An example for the analysis of BALANCE’s cost at time  $t_0$ . The line (the vertical axis) is shown in the logarithmic scale. The algorithm’s transmissions end with empty triangles. The adversary’s transmissions end with circles. The algorithm’s cost at  $t_0$  is charged to the waiting cost of packets within the shaded area and the  $2^{b-1}$ -to- $2^{b-2}$  segment of the adversarial transmission (marked with thick line) at  $t_{ADV}$ .

**Algorithm BALANCE.** At every time  $t$ , transmit from the maximum point  $2^j$  such that  $\text{twc}_t(0, 2^j] = 2^{j-2}$ , if such  $j$  exists; otherwise stay idle. (Note that every packet will eventually be transmitted.)

**Theorem 1.** BALANCE is 5-competitive.

*Proof.* Assume that BALANCE transmits from  $2^j$  at time  $t_0$ . Then, it pays  $2^j$  for the transmission and it has paid  $\text{twc}(0, 2^j] = 2^{j-2}$  for waiting of the packets it has just transmitted. Altogether, these terms contribute  $5 \cdot 2^{j-2}$  to the total cost of BALANCE. We show that it is possible to charge this cost to adversary’s actions (waiting or transmitting), whose cost is at least  $2^{j-2}$ , assuring that no action will be charged more than once. This implies that BALANCE is 5-competitive.

To this end, we choose the sequence of BALANCE’s transmissions at times  $t_1, t_2, \dots, t_k$ , going back in time, with  $k$  largest possible, where  $t_1$  is the last transmission time before  $t_0$ , each other  $t_i$  has a transmission longer than the one at  $t_{i-1}$ , and the transmission at  $t_k$  has length at least  $2^j$ . (To avoid boundary cases, we assume that there is an artificial transmission at time  $-1$  of infinite length and zero cost, so  $t_k$  is always well defined.) Formally, we find a sequence of transmission times  $t_k < t_{k-1} < \dots < t_2 < t_1 < t_0$ , such that

1.  $\text{trl}(t_k) \geq 2^j$ ,
2.  $\text{trl}(t_{i+1}) > \text{trl}(t_i)$  for  $1 \leq i \leq k-1$ ,
3.  $\text{trl}(t) \leq \text{trl}(t_i)$  for  $0 \leq i \leq k-1$  and  $t \in (t_{i+1}, t_i)$ , and  $\text{trl}(t) = 0$  for  $t \in (t_1, t_0)$ .

We call these transmissions the *cover sequence* for  $t_0$ , cf. Fig. 1.

Then, we consider adversarial transmissions occurring within the time interval  $(t_k, t_0]$ . An adversarial transmission occurring at time  $t \in (t_{\ell+1}, t_\ell]$  is called *unobstructed* if its length is at least  $\text{trl}(t_\ell)$ . Let  $t_{ADV} \in (t_k, t_0]$  be the time of the longest unobstructed adversary’s transmission (with ties broken in favor of later transmissions), and let  $A = \text{trl}_{ADV}(t_{ADV})$ ; if there was no unobstructed adversary’s transmission in  $(t_k, t_0]$ , then let  $t_{ADV} = t_0$  and  $A = 0$ . Finally, let  $b = \min\{j, \lfloor \log_2 A \rfloor + 1\}$ , where  $b = -\infty$  if  $A = 0$ . Thus, for  $A \in (0, 2^j)$ , it holds that  $2^{b-1} \leq A < 2^b$ .

Now we focus on packets at points from  $(2^b, 2^j]$  that are transmitted at  $t_0$ ; their waiting periods are contained in the shaded area in Fig. 1. Let us denote these packets by  $W(t_0)$ . Note that the waiting cost of  $W(t_0)$  is  $\text{twc}_{t_0}(2^b, 2^j]$ , and that, by the definition of  $t_{\text{ADV}}$ , the adversary pays for the waiting of these packets at least as much as BALANCE does.

We will charge BALANCE's cost of  $5 \cdot 2^{j-2}$  to two actions of the adversary: its waiting of the packets in  $W(t_0)$  and the movement of packets from  $2^{b-1}$  to  $2^{b-2}$  during the transmission at  $t_{\text{ADV}}$  (if there was any). In other words, we charge to the segment  $[2^{b-2}, 2^{b-1}]$  of the adversary's transmission at  $t_{\text{ADV}}$ . As mentioned at the beginning of the proof, it is sufficient to show two properties: (i) the total cost of these two actions is at least  $2^{j-2}$  and (ii) none of these actions is charged again when we analyze another transmission of algorithm BALANCE.

For property (i), we claim that  $\text{twc}_{t_0}(2^b, 2^j] \geq 2^{j-2} - 2^{b-2}$ . Clearly, this is the case for  $b = j$  or  $b = -\infty$ . For  $b < j$ , recall that  $\text{twc}_{t_0}(0, 2^j] = 2^{j-2}$  and furthermore,  $\text{twc}_{t_0}(0, 2^b] \leq 2^{b-2}$  as otherwise the algorithm would have transmitted from  $2^b$  earlier. The adversary cost of transmitting across the segment  $[2^{b-1}, 2^{b-2}]$  at time  $t_{\text{ADV}}$  is  $2^{b-2}$ . Together, these costs add up to at least  $2^{j-2} - 2^{b-2} + 2^{b-2} = 2^{j-2}$ , as claimed in (i).

To show property (ii), consider a transmission of BALANCE at some time  $t'_0 > t_0$  with  $\text{trl}(t'_0) = 2^{j'}$ . We also consider the corresponding cover sequence for  $t'_0$ , the time  $t'_{\text{ADV}}$  of the longest unobstructed adversarial transmission within that cover sequence, and the corresponding values of  $A' = \text{trl}_{\text{ADV}}(t'_{\text{ADV}})$  and  $b'$ .  $W(t_0)$  and  $W(t'_0)$  are disjoint sets of packets since BALANCE transmits them at distinct times  $t_0$  and  $t'_0$ . Thus, it suffices to prove that transmissions at  $t_0$  and  $t'_0$  charge their costs to different parts of the adversarial transmissions. Clearly, it is the case when they are charged to different transmissions, so in the following we assume that  $t'_{\text{ADV}} = t_{\text{ADV}}$  (and hence  $A = A'$ ). Note that  $t_{\text{ADV}} \leq t_0 < t'_0$ . Then  $j < j'$ , as otherwise the cover sequence for  $t'_0$  would end at some time point after  $t_0$ . Furthermore, the adversarial transmission at  $t_{\text{ADV}}$  is unobstructed in the cover sequence for  $t'_0$ , which means that  $A \geq 2^j$ , and hence  $b = j$ . But then, as  $j' > j$ , by the definition of  $b'$ , we have  $b' > j$ . This means that  $b' \neq b$ , i.e., the transmissions at  $t_0$  and  $t'_0$  are charged to different parts of the adversarial transmission at  $t'_{\text{ADV}} = t_{\text{ADV}}$ .  $\square$

The analysis of our algorithm can be shown to be tight. In fact, we can prove (the proof will be given in the full version) an even stronger tightness result, namely the following: *Every deterministic algorithm that transmits only from integer powers of 2 has competitive ratio at least 5.*

## 4 A Lower Bound of $2 + \phi \approx 3.618$

To prove the lower bound of  $2 + \phi$ , it is sufficient to show that for any  $R < 2 + \phi$  there is a strategy for the adversary that forces any deterministic algorithm ALG to pay at least  $R$  times the cost of an optimal solution OPT.

In the next section we show how to construct an adversarial strategy for a slightly modified version of the problem, called the *single-phase game*. In such

a game, the adversary injects (weighted) packets only at time 0, at points  $b_1 < b_2 < \dots < b_m$  that will be specified later. Additionally, the adversary has the capability to end the game at an arbitrary time  $\tau$ . As an algorithm may finish with non-transmitted packets, the definition of cost associated with such a packet  $q$  has to be adapted: it is simply the waiting time of  $q$  from time 0 till time  $\tau$ . We define *phase ratio* as the ALG-to-OPT cost ratio with the waiting costs modified as described above. Our single-phase construction has two additional properties, namely that there exist absolute lower and upper bounds on the duration of a game, and there exists an absolute lower bound on the cost of ALG in a single phase.

While the actual adversarial construction of a single-phase game and its analysis are given in the subsequent subsections, here we argue that if the adversary can force the phase ratio to be at least  $R$ , then  $R$  is a lower bound on the competitive ratio for CMA on chain networks. To this end, the adversary chooses a large integer  $\ell$ , and the actual input sequence consist of  $\ell + 1$  phases, numbered  $0, 1, \dots, \ell$ . In a phase  $p$ , the adversary plays the single-phase game, but with the weights of the packets multiplied by  $\ell^{p-\ell}$  and all the time values used in his strategy divided by  $\ell^{p-\ell}$ . Intuitively, increased weights cause waiting costs to accumulate faster, but we compensate for it by “shrinking” the time. With this rescaling, the adversarial single-phase strategy will also force ratio  $R$  in the single-phase game played in phase  $p$ . Clearly, the cost of ALG in the whole CMA instance is at least the sum of its costs in the individual single-phase games. (It could be larger if some packets are not transmitted in the phase when they are issued.) On the other hand, since the time intervals of consecutive phases decrease so fast, if a packet is injected at the beginning of phase  $p$ , and is not transmitted by the adversary within phase  $p$ , then its remaining waiting cost, in phases  $p + 1, p + 2, \dots, \ell$ , is negligible. Finally, all the packets not sent by the adversary by the end of phase  $\ell$  can be sent at that time at the cost of at most  $b_m$ , and this cost’s contribution is also negligible in comparison to the total cost. Thus, except for a negligible low-order term, the adversary’s cost is also the sum of his costs in the individual single-phase games. Therefore the overall cost of ALG is at least  $R$  times the adversary’s cost, minus a low-order term. (A similar reduction to a single-phase game was used in [3].)

The rest of this section is organized in the following way. In Section 4.1, we present the strategy of the adversary for a single phase. For the construction, the adversary has to carefully choose the number of injected packets  $m$ , their injection points  $b_1 < b_2 < \dots < b_m$ , and some waiting thresholds  $w_1, \dots, w_m$ . We list the desired properties of these sequences and show that if these properties are satisfied, then the phase ratio is at least  $R$ . Finally, in Section 4.2, we prove the existence of such sequences.

#### 4.1 Construction of a Single Phase

For the construction, we define the infinite sequences of reals  $\{b_i\}$  and  $\{w_i\}$ , where  $b_0 = w_0 = 0$  and  $b_1 = 1$ . Using notation  $B_j = \sum_{i=1}^j b_i$  and  $W_j = \sum_{i=1}^j w_i$

(thus,  $B_0 = W_0 = 0$ ), the two sequences are defined by:

$$w_j = \frac{1}{R-1} \cdot (W_{j-1} + B_j - Rb_{j-1}) \text{ for } j \geq 1, \quad (1)$$

$$b_j = R \cdot b_{j-1} + b_{j-2} - B_{j-1} - W_{j-2} \text{ for } j \geq 2. \quad (2)$$

From (2),

$$W_{j-1} = Rb_j + b_{j-1} - B_{j+1} \text{ for } j \geq 1. \quad (3)$$

Plugging this into (1) and rearranging, we obtain the following useful identity:

$$(R-1)w_j = -b_{j+1} + Rb_j - (R-1)b_{j-1} \text{ for } j \geq 1 \quad (4)$$

The crux of our construction lies in the algebraic property that, for  $R \in (2, 2+\phi)$ , the sequence  $\{b_i\}$  stops increasing at some point. In particular, in Section 4.2, we show the following crucial lemma.

**Lemma 1.** *For  $R \in (2, 2+\phi)$  and the sequences  $\{b_j\}, \{B_j\}, \{w_j\}, \{W_j\}$  defined by equations (1) and (2), there exists an integer  $m \geq 1$ , such that the following properties hold.*

- (i)  $1 = b_1 < b_2 < \dots < b_m$  and  $b_{m+1} \leq b_m$ .
- (ii)  $w_j \geq 0$  for all  $0 \leq j \leq m$ .

Note that it may happen that  $b_{m+1} < 0$ , but this does not affect the validity of our proof, since we do not use  $b_{m+1}$  as a packet injection point in the lower bound strategy; in the argument below we only need that  $b_{m+1} \leq b_m$ .

The adversary chooses  $m$ , whose existence is guaranteed by Lemma 1 and a very large integer  $K$ . At time 0 the adversary injects a packet of weight  $K^{m-j}$  at point  $b_j$ , for  $j = 1, 2, \dots, m$ .  $K$  is chosen to be at least  $\max_j(w_{j+1}/w_j)$ , which guarantees that the time when the waiting cost for the packet in  $b_j$  reaches  $w_j$  is an increasing function of  $j$ . Given that no further packets are injected within the phase, within this phase ALG will execute a sequence of transmissions from increasing points. Suppose that  $k$  is the largest integer such that ALG transmitted from  $b_k$  and  $k < m$ . Then ALG pays for the waiting cost of the packets at  $b_{k+1}, \dots, b_m$ . We will not charge ALG for the waiting cost of packets at  $b_{k+2}, \dots, b_m$ . In fact, in the calculations we will also not charge the adversary for the waiting cost of these packets. This decreases the adversary cost, but since the waiting cost of packets at  $b_{k+2}, \dots, b_m$  is negligible, this decrease is negligible as well. (A rigorous limit argument will appear in the full version of the paper.)

**The adversarial strategy.** We can assume that ALG transmits each packet if the phase is long enough, for otherwise the cost of ALG would be unbounded. With this assumption, the adversarial strategy is this: Suppose that the last transmission from ALG was from  $b_{j-1}$  and let  $\omega$  be the waiting cost of the packets in  $b_j$  when ALG makes the next transmission, from some  $b_{j'}$  (where  $j' \geq j$ ). Then, if  $j' = j < m$  and  $\omega \geq w_j$ , the adversary continues the phase. Otherwise, the phase ends.

The intuition is that the adversary tries to force ALG to transmit from points  $b_1, b_2, \dots$ , one by one. Suppose that the adversary ends the game at step  $j$ . The algorithm paid both the waiting cost and the transmission cost of the packets at  $b_1, \dots, b_{j-1}$ , while the adversary can perform significantly better by transmitting from  $b_{j-1}$  at the beginning of the phase and not paying their waiting cost at all. If ALG skips  $b_j$  and transmits from  $b_{j'}$ , for  $j' > j$ , then it pays extra transmission cost. On the other hand, if it transmits from  $b_j$  before the threshold  $w_j$ , the phase will be short, so the adversary can save cost by not transmitting from  $b_j$ . The formal argument follows the theorem below.

**Theorem 2.** *Every deterministic algorithm has competitive ratio at least  $2 + \phi$ .*

*Proof.* As explained at the beginning of Section 4, it is sufficient to show that for any  $R \in (2, 2 + \phi)$ , the phase ratio is at least  $R$ . The adversary uses the strategy described above.

Suppose that the phase ends at step  $j$ . Then, up to this step, ALG made separate transmissions from  $b_1, b_2, \dots, b_{j-1}$ , and for each of these  $b_i$  the waiting cost of the packet at  $b_i$  was at least  $w_i$ . So the cost of ALG so far is at least  $W_{j-1} + B_{j-1}$ , plus the cost associated with the packet at  $b_j$ . For determining the phase ratio, we consider three possibilities of finishing the phase. Notice that, by Lemma 1, in each case both the numerator and the denominator are positive. Recall also that, as explained earlier, we can ignore the waiting costs for packets at  $b_{j+1}, \dots, b_m$ .

Case 1:  $j' > j$ . Note that this implies  $j < m$ . ALG's cost is  $W_{j-1} + B_{j-1} + \omega + b_{j'}$ . We consider two options for the adversary: he can transmit from  $b_{j-1}$  at time 0 and pay  $\omega$  for waiting, or he can transmit from  $b_j$ . So the ratio is

$$\begin{aligned} \frac{W_{j-1} + B_{j-1} + \omega + b_{j'}}{\min(b_{j-1} + \omega, b_j)} &\geq \frac{W_{j-1} + B_{j-1} + \omega + b_{j+1}}{\min(b_{j-1} + \omega, b_j)} \\ &\geq \frac{1}{b_j} (W_{j-1} + B_{j-1} + b_j - b_{j-1} + b_{j+1}) \\ &= \frac{1}{b_j} (W_{j-1} + B_{j+1} - b_{j-1}) \\ &= \frac{1}{b_j} (b_{j+1} - b_{j-1} + B_j + W_{j-1}) = R, \end{aligned}$$

where the last equality follows from (2).

Case 2:  $j' = j$  and  $\omega < w_j$ . ALG's cost is at least  $W_{j-1} + B_j + \omega$ . The adversary will transmit from  $b_{j-1}$  at time 0 and pay the cost of waiting at  $b_j$ , paying  $b_{j-1} + \omega$ . So the ratio is

$$\frac{W_{j-1} + B_j + \omega}{b_{j-1} + \omega} \geq \frac{W_{j-1} + B_j + w_j}{b_{j-1} + w_j} = R,$$

where the last equality follows from (1).

Case 3:  $j' = j = m$  and  $\omega \geq w_j$ . The cost of ALG is  $W_{m-1} + B_m + \omega$ . The adversary will transmit from  $b_m$  at time 0, paying  $b_m$ . Note that, by the choice of  $m$  we have  $b_{m+1} \leq b_m$ . By (4),  $(R-1)w_m = -b_{m+1} + Rb_m - (R-1)b_{m-1} \geq (R-1)(b_m - b_{m-1})$ , that is  $w_m \geq b_m - b_{m-1}$ . So the ratio is

$$\frac{W_{m-1} + B_m + \omega}{b_m} \geq \frac{W_{m-1} + B_m + w_m}{b_{m-1} + w_m} = R,$$

where the last equality again follows from (1).  $\square$

## 4.2 Proof of Lemma 1

**Proof of Part (i).** We first derive a single recurrence for the sequence  $\{b_j\}$ . Assuming  $j \geq 1$ , plugging (3) into (2) and simplifying, we obtain the recurrence

$$(R-1)b_{j+2} - (R^2 - R + 1)b_{j+1} + (R^2 - R + 1)b_j = 0 \quad \text{for } j \geq 1. \quad (5)$$

with  $b_0 = 0$ ,  $b_1 = 1$  and  $b_2 = R - 1$ . (Note that an initial condition for  $j = 2$  is not covered by (5).) The characteristic equation of (5) is  $(R-1)x^2 - (R^2 - R + 1)x + (R^2 - R + 1) = 0$  with discriminant  $\Delta = (R^2 - R + 1)(R^2 - 5R + 5)$ . In the interval  $(2, 2 + \phi)$  the value of  $\Delta$  is negative, so the characteristic equation has two imaginary conjugate roots

$$\beta_{1,2} = \frac{R^2 - R + 1 \pm \sqrt{\Delta}}{2(R-1)},$$

and consequently

$$b_j = \alpha_1 \cdot \beta_1^j + \alpha_2 \cdot \beta_2^j, \quad j \geq 1, \quad (6)$$

for some complex numbers  $\alpha_1, \alpha_2 \neq 0$ . From the theory of recurrence equations of order 2 (the case of conjugate imaginary roots), the sequence of  $\{b_j\}$  cannot increase infinitely (cf. Appendix A), which proves Part (i) of Lemma 1.

**Proof of Part (ii).** It is sufficient to show that the right hand side of (4) is non-negative, i.e., that

$$-b_{j+1} + Rb_j - (R-1)b_{j-1} \geq 0 \quad \text{for } j \geq 1. \quad (7)$$

For  $j = 1$ , the left-hand-side of (7) is  $-b_2 + Rb_1 - (R-1)b_0 = -(R-1) + R \cdot 1 - (R-1) \cdot 0 = 1 \geq 0$ . Suppose  $j \geq 2$  and that the claim holds for  $j-1$ . After multiplying the left-hand side of (7) by  $R-1$  and rearranging it, we get

$$\begin{aligned} & (R-1)[-b_{j+1} + Rb_j - (R-1)b_{j-1}] \\ &= [-(R-1)b_{j+1} + (R^2 - R + 1)b_j - (R^2 - R + 1)b_{j-1}] \\ & \quad + [-b_j + Rb_{j-1} - (R-1)b_{j-2}] + (R-1)b_{j-2} \geq 0, \end{aligned}$$

The last inequality follows because the first term is 0 by (5), the second one is non-negative by the inductive assumption, and the third one is positive by the choice of  $m$ .

## 5 Polynomial-Time Offline Solution

The input is a sequence of packets numbered  $1, 2, \dots, n$ , where packet  $j$  is specified as a triple  $(t_j, x_j, w_j)$ . In this triple,  $t_j$  is the injection time,  $x_j$  is the point of injection, and  $w_j$  is the weight of packet  $j$ . For simplicity, we assume that  $t_1 < t_2 < \dots < t_n$  and that all  $x_j$  are different. Any instance can be modified to have this form by infinitesimal perturbations on the time and space axes.

Without loss of generality, we can assume that in an optimal solution each transmission occurs at some time  $t_k$  and it includes the packet injected at this time, that is, it transmits from some point  $x_j$  where  $x_j \geq x_k$ . We can further assume that  $j \leq k$ , since otherwise  $x_j$  itself does not yet have a packet. We call a transmission from  $x_j$  at time  $t_k$  satisfying these conditions a  $(j, k)$ -*transmission*.

For  $i < k$  and any  $j$  we define a sub-instance  $\mathcal{I}_{i,j,k}$  that consists of the triples  $(t_a, x_a, w_a)$  such that  $i < a < k$  and  $x_a < x_j$ . We consider the quantity  $F_{i,j,k}$  that represents the minimum cost of sub-instance  $\mathcal{I}_{i,j,k}$  accrued in the time window  $(t_i, t_k)$ . To define it formally, we relax the rules to allow some packets in  $\mathcal{I}_{i,j,k}$  not to be transmitted. The cost of transmissions is defined as before. The waiting cost of each packet is either the cost of waiting until its transmission, if it gets transmitted, or until time  $t_k$ , if it's not. Then  $F_{i,j,k}$  is the minimum cost of  $\mathcal{I}_{i,j,k}$  under these rules.

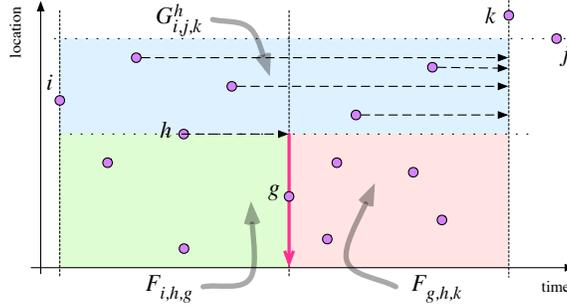
We now derive the recurrence equation for the  $F_{i,j,k}$ 's. For  $k = i+1$ ,  $\mathcal{I}_{i,j,k} = \emptyset$ , so we have  $F_{i,j,k} = 0$ . Let  $k > i+1$ . For any  $h$ , let  $G_{i,j,k}^h$  be the waiting time for packets from  $\mathcal{I}_{i,j,k}$  that are beyond  $x_h$ , assuming that they are not transmitted before  $t_k$ , cf. Figure 2. Thus

$$G_{i,j,k}^h = \sum_{\substack{i < \ell < k \\ x_h < x_\ell < x_j}} w_\ell (t_k - t_\ell).$$

Slightly abusing notation, we will allow  $h = 0$  in the above formula, with  $x_0$  understood to be 0. Thus  $G_{i,j,k}^0$  is the total waiting cost of packets in  $\mathcal{I}_{i,j,k}$  if there are no transmissions. We then claim that

$$F_{i,j,k} = \min \left\{ \begin{array}{l} G_{i,j,k}^0 \\ \min_{\substack{i < h \leq g < k \\ x_g \leq x_h < x_j}} \{ F_{i,h,g} + x_h + w_h(t_g - t_h) + F_{g,h,k} + G_{i,j,k}^h \} \end{array} \right.$$

The recurrence is illustrated in Fig. 2. To show correctness, we argue as follows. Consider the optimum schedule for  $\mathcal{I}_{i,j,k}$ . If no transmissions occur in this schedule, then  $F_{i,j,k} = G_{i,j,k}^0$ . If there is at least one transmission, choose the  $(h, g)$ -transmission with maximum  $x_h$ . Then all packets  $\ell$  in  $\mathcal{I}_{i,j,k}$  above  $x_h$  pay the waiting cost for the interval  $[t_\ell, t_k]$ . The total of these waiting costs is  $G_{i,j,k}^h$ , the last term in the formula. The cost of the  $(h, g)$ -transmission is  $x_h$  and we need to pay for the waiting cost of  $x_h$ , which is equal  $w_h(t_g - t_h)$ . These are the second and third terms in the formula. We then need to add the cost of serving the packets of  $\mathcal{I}_{i,j,k}$  that are below  $x_h$ . These packets constitute two



**Fig. 2.** The idea behind the recurrence for  $F_{i,j,k}$ .

sub-instances,  $\mathcal{I}_{i,h,g}$  and  $\mathcal{I}_{g,h,k}$ , with respective costs  $F_{i,h,g}$  and  $F_{g,h,k}$ , which are the first and the fourth terms in the formula.

It remains to show how to use the recurrence to compute the solution of the whole instance. To this end, we modify the instance by adding two new packets of weight 0, injected first and last. Denoting by  $n$  the number of packets in this new instance, the original instance will consist of packets injected into  $x_2, \dots, x_{n-1}$  at times  $t_2, \dots, t_{n-1}$ . We set  $t_1 < t_2$ ,  $t_n > t_{n-1}$  with the difference  $t_n - t_{n-1}$  large enough, so that in the optimum solution no packets from the original instance will wait until time  $t_n$ . Also,  $x_1 < x_n$  are both larger than all  $x_2, \dots, x_{n-1}$ . Then,  $F_{1,n,n}$  is the same as the cost of the optimum solution of the original instance.

Summarizing, we proved the following result:

**Theorem 3.** *The optimum solution for message aggregation on chain networks can be computed in time  $O(n^5)$ .*

## 6 Final Comments

For the online Control Message Aggregation problem on chain networks we proved that the optimal competitive ratio is between 3.681 and 5. Closing or tightening this gap remains an open problem, although we showed that to improve the upper bound a new approach is needed. We have not addressed the case of randomized algorithms and we leave it for future work. It is intuitively clear that randomization should help to reduce the upper bound. In fact, there are at least two ways to take advantage of randomization: one, by choosing the cutoff points (a sequence other than the powers of 2) randomly, and two, by choosing the transmissions at random.

## References

1. Esther Arkin, Dev Joneja, and Robin Roundy. Computational complexity of uncapacitated multi-echelon production planning problems. *Operations Research Letters*, 8(2):61–66, 1989.
2. Carlos Brito, Elias Koutsoupias, and Shailesh Vaya. Competitive analysis of organization networks or multicast acknowledgement: How much to wait? *Algorithmica*, 64(4):584–605, 2012.

3. Niv Buchbinder, Tracy Kimbrel, Retsef Levi, Konstantin Makarychev, and Maxim Sviridenko. Online make-to-order joint replenishment model: primal dual competitive algorithms. In *Proc. of the 19th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 952–961, 2008.
4. Daniel R. Dooley, Sally A. Goldman, and Stephen D. Scott. On-line analysis of the TCP acknowledgment delay problem. *Journal of the ACM*, 48(2):243–273, 2001.
5. Anna R. Karlin, Claire Kenyon, and Dana Randall. Dynamic TCP acknowledgement and other stories about  $e/(e - 1)$ . *Algorithmica*, 36(3):209–224, 2003.
6. Sanjeev Khanna, Joseph Naor, and Danny Raz. Control message aggregation in group communication protocols. In *Proc. of the 29th Int. Colloq. on Automata, Languages and Programming (ICALP)*, pages 135–146, 2002.
7. Retsef Levi, Robin Roundy, and David B. Shmoys. A constant approximation algorithm for the one-warehouse multi-retailer problem. In *Proc. of the 16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 365–374, 2005.
8. Retsef Levi, Robin Roundy, David B. Shmoys, and Maxim Sviridenko. A constant approximation algorithm for the one-warehouse multiretailer problem. *Management Science*, 54(4):763–776, 2008.
9. Retsef Levi and Maxim Sviridenko. Improved approximation algorithm for the one-warehouse multi-retailer problem. In *Proc. of the 9th Int. Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, pages 188–199, 2006.

## A Order 2 Recurrence Equations with Non-Real Base Solutions

For completeness, we prove here that for every real-valued sequence  $\{b_i\}$  defined by a recurrence equation of order 2 with non-real base solutions, there exists  $n > 1$  such that  $b_n \leq 0$ . Note that for the sequence from our lower bound construction, this trivially implies the existence of  $m$ .

Let  $\alpha_1, \alpha_2, \beta_1, \beta_2$  be as defined in (6), and let us assume for convenience that (6) holds for  $j = 0$  as well. Since  $\beta_1$  and  $\beta_2$  are non-real roots of a quadratic equation, it follows that  $\beta_2 = \overline{\beta_1}$ , i.e., they are complex conjugates.

In the following, we reason about the imaginary part of (6), which is 0 for all  $j \in \mathbb{N}$  by our assumption. Laying  $j = 0$ , we have

$$\Im(\alpha_1) + \Im(\alpha_2) = 0 \quad , \quad (8)$$

and laying  $j = 1$ , we have

$$\Im(\alpha_1) \cdot \Re(\beta_1) + \Re(\alpha_1) \cdot \Im(\beta_1) + \Im(\alpha_2) \cdot \Re(\beta_2) + \Re(\alpha_2) \cdot \Im(\beta_2) = 0 \quad . \quad (9)$$

As  $\beta_2 = \overline{\beta_1}$ , we have  $\Re(\beta_1) = \Re(\beta_2)$ . Together with (8) this implies  $\Im(\alpha_1) \cdot \Re(\beta_1) + \Im(\alpha_2) \cdot \Re(\beta_2) = 0$ , which subtracted from (9) yields

$$\Re(\alpha_1) \cdot \Im(\beta_1) + \Re(\alpha_2) \cdot \Im(\beta_2) = 0 \quad .$$

As  $\Im(\beta_1) + \Im(\beta_2) = 0$ , this implies  $\Re(\alpha_1) = \Re(\alpha_2)$ , which together with (8) means that  $\alpha_2 = \overline{\alpha_1}$ .

Thus to get  $b_n \leq 0$ , it suffices to pick  $n$  such that  $\arg(\alpha_1 \cdot \beta_1^n) \in [\pi/2, 3\pi/2]$ , which is equivalent to  $\arg(\alpha_1) + n \cdot \arg(\beta_1) \in [\pi/2 + 2k\pi, 3\pi/2 + 2k\pi]$  for some  $k \in \mathbb{N}$ . This is possible, since  $\beta_1 \notin \mathbb{R}$  means that  $\arg(\beta_1)$  is not an integer multiple of  $\pi$ .