New Lower and Upper Bounds for the Competitive Ratio of Transmission Protocols

Maciej Liśkiewicz [1], Bodo Manthey [2]

Universität zu Lübeck, Institut für Theoretische Informatik
Wallstraße 40, 23560 Lübeck, Germany

Abstract

Transmission protocols like TCP are usually divided into a time scheduling and a data selection policy. We consider on-line algorithms of data selection policies for any time scheduling policy and any routing behaviour in a network. For the model introduced by Adler et al. (1997, Proc. of the 5th Isreal Symp. on the Theory of Computing Systems, pp. 64–72), we improve both the lower and the upper bound on the competitive ratio making them asymptotically tight. Furthermore, we present a lower bound that depends on the size of the buffers that are available both to the sender and to the receiver. We obtain a constant lower bound for the competitive ratio for constant buffer size.

Key words: on-line algorithms, interconnection networks, transmission protocols

1 Introduction

We consider protocols for transmitting data through a communication network like the Internet. Data are transmitted in chunks called packets and we allow that the network can lose transmitted packets. To construct such protocols one uses a modular approach (see Clark et al. [5]). According to Mathis and Mahdavi [6] one can decompose the protocols into a *time scheduling policy* that decides *when* to send a packet and *data selection policy* that decides *which* piece of data should be transmitted with a particular packet. For our investigation, we use the model of transmission protocols proposed by Adler

Email addresses: `liskiewi@tcs.uni-luebeck.de` (Maciej Liśkiewicz), `manthey@tcs.uni-luebeck.de` (Bodo Manthey).
1 On leave from Instytut Informatyki, Uniwersytet Wrocławski, Poland.
2 Supported by DFG research grant Re 672/3.

et al. [1] and Byers [4] which allows the analysis of data selection policies for any time scheduling policy and any behaviour of a network.

1.1 The Model

We study the point-to-point communication, where a sender transmits typically huge data to a single receiver. The model that we use is based on developments of Adler et al. [1] and Byers [4]. It has further been applied by Bartal et al. [2] to analyse performance of protocols for multicast connections, too.

The sender can send packets, where packet p_i contains a piece of data that we call a *word*. Furthermore, the sender receives positive and negative acknowledgements that tell him whether some packet has successfully been transmitted or has been lost. We say that a packet has been *accepted* if the sender has received a positive acknowledgement for it. Similarly, we say that a packet has been *rejected* in case of a negative acknowledgement. Analogously, a word has been accepted if there is one packet containing this word that has been accepted.

We use the following notation based on Adler et al. [1]:

- s_i is the time at which packet p_i has been transmitted,
- a_i is the time at which the sender gets an acknowledgement for packet p_i,
- b_i is 1, if packet p_i has been transmitted successfully, and 0 otherwise,
- $a_i - s_i$ is the round-trip time of packet p_i, and
- w_i is the data of packet p_i, i.e., the word transmitted using packet p_i.

We call the collection of 4-tuples $(s_i, a_i, b_i, w_i)_{i\in\mathbb{N}}$ for all packets *transcript* θ. The *backlog* $B(t,\theta)$ for a transcript θ at a time t is the number of packets for which the sender has no acknowledgement yet. B_θ is the backlog of θ, i.e., $B_\theta = \max_{t\in\mathbb{N}} B(t,\theta)$. The available *bandwidth* until time t with transcript θ is

$$P^\star(t,\theta) = \sum_{i \text{ with } a_i \le t} b_i \,,$$

i.e., the number of positive acknowledgements received by the sender until time t. Let A be a data selection policy, i.e., an on-line algorithm that decides which word to put into which packet and let $u_1, u_2, u_3, \ldots$ be a data stream the sender wants to transmit. As quality measure we consider the length of the longest prefix that is accepted. The maximum prefix length $P^A(t,\theta)$ is the largest integer j such that all words $u_1, \ldots, u_j$ have already been accepted prior to time t.

An optimal data selection policy has successfully transmitted the first $P^\star(t,\theta)$

words until time t. We define the *competitive ratio* of an algorithm A in a slightly different manner than usual (see e.g. Borodin and El-Yaniv [3]):

$$R^A(t,\theta) = \frac{P^\star(t,\theta)}{P^A(t,\theta)} .$$

The aim is to find an algorithm that keeps the competitive ratio as small as possible.

Another aim is to guarantee that the buffer protocol A needs does not become too large. Let $S^A(t,\theta)$ denote the *size of the buffer* A needs at time t with transcript θ, i.e., $S^A(t,\theta)$ is the maximum number of different words sent by A prior to time t that do not belong to the longest contiguous prefix transmitted successfully. Speaking more formally, if $I^A(t,\theta)$ is the largest index of a word from the input data stream $u_1, u_2, u_3, \ldots$ that has been sent by A prior to t, then we have

$$S^A(t,\theta) = \max_{0<t'\leq t} \left\{ I^A(t',\theta) - P^A(t',\theta) \right\} .$$

1.2 Previous Results

Adler et al. [1] have presented the following lower and upper bounds on the competitive ratio for deterministic data selection policies.

Theorem 1 (Upper Bound [1, Thm. 1]) *There exists a deterministic algorithm A, such that for all transcripts θ and for all times t,*

$$R^A(t,\theta) = 1 + O\left(\frac{(B_\theta \log B_\theta)^{\frac{3}{2}}}{\sqrt{P^\star(t,\theta)}} \right) .$$

Theorem 2 (Lower Bound [1, Thm. 2]) *For any deterministic algorithm A, there exist a transcript θ and a time t, for which*

$$R^A(t,\theta) = 1 + \Omega\left(\left(\sqrt{P^\star(t,\theta)} \right)^{-1} \right) .$$

Adler et al. use the following adversarial strategy $\mathcal{S}$ to prove Theorem 2: Repeatedly allow the algorithm to transmit two packets. If they contain identical words, then accept both, otherwise accept only the packet containing the word with the larger index. Then they claim [1, Claim 6]: *For any deterministic data selection policy A, and any transcript θ resulting from the adversarial strategy $\mathcal{S}$, there exists a time t in which $R^A(t,\theta) \geq 1 + \frac{1}{2\sqrt{P^\star(t,\theta)}}$.*

We find that this claim does not hold if we allow to send words that have been accepted before. Consider the following algorithm. First send word u_1 twice using packets p_1 and p_2. Then send u_i and again u_1 using the packets p_{2i-1} and p_{2i}. Thus for $t \geq 5$, we have $R^A(t,\theta) = \frac{P^\star(t,\theta)+1}{P^\star(t,\theta)} < 1 + \frac{1}{2\sqrt{P^\star(t,\theta)}}$.

We can avoid this problem by slightly modifying the adversary: If one of the two words transmitted has been accepted previously, then accept this packet and reject the other. We call the modified adversary $\mathcal{S}'$.

1.3 Our Results

We improve both the lower and upper bound for the competitive ratio of deterministic data selection policies. Our bounds are asymptotically tight. Furthermore, we extend the analysis by taking the available buffer into account. We show a lower bound for the competitive ratio of any deterministic data selection policy that depends on the buffer size available. In particular, we obtain a constant lower bound for constant buffer size.

2 New Lower and Upper Bounds

In this section we show improvements of the Theorems 1 and 2. Our first result directly improves Theorem 2: We show that for any algorithm there exists a transcript θ and a time t in which the competitive ratio exceeds the lower bound. This lower bound is asymptotically tight to the upper bound given by Adler et al. [1]. However, a much more interesting property is to find a lower bound that is exceeded for infinitely many times. In fact, the proof of Theorem 2 given by Adler et al. [1] holds also for infinitely many times. Theorem 4 below, improves on their result. On the other hand, we present a protocol (Theorem 6) that achieves after a certain time a competitive ratio that is asymptotically tight to the lower bound given in Theorem 4. This protocol needs to know the backlog in advance. We generalize this result in Corollary 7. Note that this does not contradict Proposition 3, although Proposition 3 holds even if we know the backlog in advance.

Proposition 3 *For every deterministic algorithm A, there exists a transcript θ and a time t for which*

$$R^A(t,\theta) \geq 1 + \frac{B_\theta^{\frac{3}{2}}}{36\sqrt{2 \cdot P^\star(t,\theta)}} .$$

PROOF. Let A be a transmission protocol. We define a transcript θ with an arbitrary backlog $B_\theta \geq 3$ as follows. Let $s_i = i$ and $a_i = i + B_\theta$ for all $i \in \mathbb{N}$. To determine the b_i-values for $1 \leq i \leq B_\theta$, we consider two cases:

- If the first word u_1 of the input data stream occurs at least $B_\theta/2$ times in the sequence $w_1, \ldots, w_{B_\theta}$, then accept all packets containing u_1 and reject all other packets.
- If the word u_1 occurs less than $B_\theta/2$ times in $w_1, \ldots, w_{B_\theta}$, then reject all packets containing u_1 and accept all other packets.

We do not specify the transcript for $p_{B_\theta+1}, \ldots, p_{2B_\theta}$. In time $t = 2B_\theta$ we have acknowledgements for the first B_θ packets. Thus, $P^A(t,\theta) \leq 1$ and $B_\theta/2 \leq P^\star(t,\theta) \leq B_\theta$. Hence, we have $R^A(t,\theta) \geq B_\theta/2 \geq B_\theta^{\frac{3}{2}} / \left(2\sqrt{2P^\star(t,\theta)}\right) \geq 1 + B_\theta^{\frac{3}{2}} / \left(36\sqrt{2 \cdot P^\star(t,\theta)}\right)$. □

Theorem 4 *For every deterministic algorithm A, there exists a transcript θ and a sequence of times $(t_k)_{k\in\mathbb{N}}$ (with $t_{k+1} > t_k$) such that for all $k \in \mathbb{N}$*

$$R^A(t_k,\theta) \geq 1 + \sqrt{\frac{B_\theta\,(\ln B_\theta - 1)}{10 \cdot P^\star(t_k,\theta)}}\,.$$

PROOF. We define a transcript θ with an arbitrary backlog $B_\theta \geq 2$. First let $s_i = i$ and $a_i = i + B_\theta$. Now let A be an arbitrary deterministic algorithm. We divide the work of A into phases. At the beginning of phase n let $j_1, j_2, \ldots, j_{B_\theta}$ be the B_θ smallest indices of words that have not been accepted when phase $n-1$ was finished. Let $U_n = \{u_{j_1}, u_{j_2}, \ldots, u_{j_{B_\theta}}\}$.

In the first phase, let $j_i = i$. Phase n is finished when all words in U_n have been accepted. Now let us consider blocks of B_θ consecutive packets sent by the transcript θ. For each such block $p_{i+1}, p_{i+2}, \ldots, p_{i+B_\theta}$, the adversary considers two cases to determine the values for b_{i+j}:

- If $w_{i+j} \notin U_n$ for some $1 \leq j \leq B_\theta$, then accept every such packet and reject all other packets, i.e., all packets containing a word in U_n.
- If all $w_{i+j} \in U_n$ $(1 \leq j \leq B_\theta)$, then choose the word w that occurs most frequently in $w_{j+1}, w_{j+2}, \ldots, w_{j+B_\theta}$ (breaking ties arbitrarily), accept all packets containing w, reject all other packets, and remove w from U_n.

Claim 5 *For every phase n, the total number of positive acknowledgements for words in U_n is at least $B_\theta \cdot H_{B_\theta}$, where H_{B_θ} is the B_θth harmonic number.*

PROOF. If there is a packet in a block $p_{i+1}, p_{i+2}, \ldots, p_{i+B_\theta}$ containing a word $w_{i+j} \notin U_n$, then no word from U_n will be accepted. On the other hand, if all words w_{i+j} $(1 \le j \le B_\theta)$ belong to U_n, then the adversarial strategy enforces completing exactly one word of U_n. Hence, in phase n there are exactly B_θ blocks containing only words from U_n. Moreover, by the pigeon hole principle we can conclude that when such a block has been sent in a phase for the jth time $(1 \le j \le B_\theta)$, then there are at least $\left\lceil \frac{B_\theta}{B_\theta - j + 1} \right\rceil$ packets in the block containing the same word. Thus, the total number of positive acknowledgements is at least $\sum_{j=1}^{B_\theta} \lceil \frac{B_\theta}{B_\theta - j+1} \rceil = \sum_{j=1}^{B_\theta} \lceil \frac{B_\theta}{j} \rceil \ge B_\theta \cdot H_{B_\theta}$. □

Now we can complete the proof of the theorem. We choose an arbitrary but sufficiently large $k \in \mathbb{N}$ and t such that $P^\star(t,\theta) = \lfloor (B_\theta \cdot H_{B_\theta})^{2k} \rfloor$. We distinguish two cases. First, assume that there exists a time subinterval $[t', t'']$ $(1 \le t' < t'' \le t)$ during which at least $\left\lfloor (B_\theta \cdot H_{B_\theta})^{k+\frac{1}{2}} \right\rfloor$ packets have been accepted but no phase has been finished. Then $P^\star(t'',\theta) - P^A(t'',\theta) \ge \lfloor (B_\theta \cdot H_{B_\theta})^{k+\frac{1}{2}} \rfloor - B_\theta$ and $P^\star(t'',\theta) \le (B_\theta \cdot H_{B_\theta})^{2k}$. Thus, we obtain

$$\begin{aligned} R^A(t'',\theta) &\ge \frac{P^A(t'',\theta) + \lfloor (B_\theta \cdot H_{B_\theta})^{k+\frac{1}{2}} \rfloor - B_\theta}{P^A(t'',\theta)} \\ &\ge 1 + \frac{\lfloor (B_\theta \cdot H_{B_\theta})^{k+\frac{1}{2}} \rfloor - B_\theta}{P^\star(t'',\theta)} \ge 1 + \sqrt{\frac{B_\theta \cdot H_{B_\theta}}{10 \cdot P^\star(t'',\theta)}} \,. \end{aligned}$$

Second, if for each subinterval $[t', t'']$ during which $\left\lfloor (B_\theta \cdot H_{B_\theta})^{k+\frac{1}{2}} \right\rfloor$ packets have been accepted at least one phase has been completed, then, since there are at least $\left\lfloor (B_\theta \cdot H_{B_\theta})^{k-\frac{1}{2}} \right\rfloor$ disjoint subintervals and by Claim 5, we have $P^\star(t,\theta) - P^A(t,\theta) \ge \lfloor (B_\theta \cdot H_{B_\theta})^{k-\frac{1}{2}} \rfloor \cdot (B_\theta \cdot H_{B_\theta} - B_\theta)$ and

$$R^A(t,\theta) \ge 1 + \frac{((B_\theta H_{B_\theta})^{k-\frac{1}{2}} - 1) \cdot (B_\theta H_{B_\theta} - B_\theta)}{P^\star(t,\theta)} \ge 1 + \sqrt{\frac{B_\theta H_{B_\theta}}{10 \cdot P^\star(t,\theta)}} \,.$$

Thus, there are infinitely many k for which we get different times that fulfil the bound. Since $H_{B_\theta} \ge \ln B_\theta - 1$ for all $B \in \mathbb{N}$, the theorem is proved. □

Let us now focus on upper bounds.

Theorem 6 *There exists an algorithm A such that if we know B_θ in advance, then A achieves for sufficiently large t*

$$R^A(t,\theta) \le 1 + 4\sqrt{\frac{B_\theta \log B_\theta}{P^\star(t,\theta)}} \,.$$

PROOF. The algorithm A is similar to the deterministic algorithm presented by Adler et al. [1, Sec. 3.1]. It works in phases. In phase n we transmit the next $k_n = nB_\theta \log B_\theta$ words (*block* n) until all of them have been accepted. Then we proceed with phase $n+1$. When we are allowed to transmit a packet, we choose a word of block n for which the number of acknowledgements that we are waiting for is minimum (breaking ties arbitrarily). Adler et al. have shown that during phase n, an optimal protocol transmits at most $k_n + B_\theta \log B_\theta$ words.

Let t be a time within some phase $n+1$. Then we have $P^\star(t,\theta) \le \sum_{i=1}^{n+1}(k_i + B_\theta \log B_\theta)$ and $P^A(t,\theta) \ge \sum_{i=1}^{n} k_i$. Thus,

$$R^A(t,\theta) \le \frac{\sum_{i=1}^{n+1}(k_i + B_\theta \log B_\theta)}{\sum_{i=1}^{n} k_i} \le 1 + \frac{2(n+1)B_\theta \log B_\theta}{B_\theta(\log B_\theta)n(n+1)/2} = 1 + \frac{4}{n} .$$

Furthermore, we have $P^\star(t,\theta) \le \frac{(n+1)(n+2)}{2} B_\theta \log B_\theta + (n+1)B_\theta \log B_\theta$ and thus for $n \ge 6$, $n^2 \ge \frac{P^\star(t,\theta)}{B_\theta \log B_\theta}$. Thus for every t in phase n ($n \ge 6$), we have $R^A(t,\theta) \le 1 + 4\sqrt{\frac{B_\theta \log B_\theta}{P^\star(t,\theta)}}$. □

If we do not know the backlog in advance, we still can achieve the upper bound of the previous theorem asymptotically after a certain time. This will be done by estimating the backlog. We set $k_n = nB_n \log B_n$, where B_n is the largest backlog observed during the previous phases.

Corollary 7 *There exists a deterministic algorithm A with the following property: For all transcripts θ for which B_θ is bounded, there exists a time $t_\theta \in \mathbb{N}$ such that for all $t \ge t_\theta$ we have*

$$R^A(t,\theta) \le 1 + O\left(\sqrt{\frac{B_\theta \log B_\theta}{P^\star(t,\theta)}}\right) .$$

3 Buffer Size and Competitive Ratio

In this section, we revisit the modified adversary considered in Section 1.2 to prove a lower bound for the competitive ratio that depends on the buffer size available.

Theorem 8 *Let A be any deterministic data selection policy and θ be the transcript resulting from the adversary $\mathcal{S}'$ against protocol A. Then there exists a sequence $(t_k)_{k\in\mathbb{N}}$ of times (with $t_k < t_{i+k}$) such that $R^A(t_k,\theta) \ge 1 + \frac{1}{S^A(t_k,\theta)+1}$ for all $k \in \mathbb{N}$.*

PROOF. Consider a sequence $t_0 = 0, t_1, t_2, \ldots$ with $S^A(t,\theta) \leq \sum_{j=t_{k-1}+1}^{t_k} b_j \leq S^A(t,\theta) + 1$ for every $k = 1, 2, \ldots$. Let $\tau = t_n$. During an arbitrary interval $[t_{k-1}, t_k]$ at least $S^A(\tau,\theta)$ packets will be accepted. We claim that among all pairs of packets sent during the interval there exists at least one accepted packet containing a word that has been accepted while sent using another packet until t_k. If this would not be the case, then each pair of packets sent during the interval got exactly one positive and one negative acknowledgement. Hence, all words contained in packets that have been accepted are pairwise different. Let j_{min} be the minimum over all indices of words that have been rejected during the phase and j_{max} be the maximum over all indices of words that have been accepted during the phase. Then we have $j_{max} - j_{min} > S^A(t,\theta)$, a contradiction.

Hence, in all intervals considered, there is a packet containing a word that has been accepted while sent previously in some other packet. From the definition of R^A we have $R^A(t,\theta) = 1 + \frac{P^\star(t,\theta) - P^A(t,\theta)}{P^A(t,\theta)} \geq 1 + \frac{P^\star(t,\theta) - P^A(t,\theta)}{P^\star(t,\theta)}$. Since there are n time intervals until τ, we have $R^A(\tau,\theta) \geq 1 + \frac{n}{n(S^A(\tau,\theta)+1)} \geq 1 + \frac{1}{S^A(\tau,\theta)+1}$. □

In particular, we obtain a constant lower bound for the competitive ratio, if the buffer has constant size.

Acknowledgements

We thank the anonymous referee for remarks that helped improving the presentation.

References

[1] M. Adler, Y. Bartal, J. W. Byers, M. Luby, D. Raz, A modular analysis of network transmission protocols, in: Proc. of the 5th Israel Symp. on the Theory of Computing Systems (ISTCS), 1997, pp. 54–62.

[2] Y. Bartal, J. W. Byers, M. Luby, D. Raz, Feedback-free multicast prefix protocols, in: Proc. of the 3rd IEEE Symp. on Computers and Communication (ISCC), 1998, pp. 135–141.

[3] A. Borodin, R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998.

[4] J. W. Byers, Maximizing throughput of reliable bulk network transmissions, Ph.D. thesis, UC Berkeley (1997).

[5] D. D. Clark, M. L. Lambert, L. Zhang, NETBLT: A high throughput transport protocol, ACM SIGCOMM Computer Communication Review 17 (5) (1987) 353–359.

[6] M. Mathis, J. Mahdavi, Forward acknowledgement: Refining TCP congestion control, ACM SIGCOMM Computer Communication Review 26 (4) (1996) 281–291.