On Time Lookahead Algorithms for the Online Data Acknowledgement Problem*

Csanád Imreh[1] and Tamás Németh[2]

[1] Department of Informatics, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary
cimreh@inf.u-szeged.hu
[2] Móra Ferenc Highschool, Kálvária sgt. 84, H-6724, Szeged, Hungary
tom@moraisk.hu

Abstract. In this work we investigate such online algorithms for the data acknowledgement problem, which have extra information about the arrival time of the packets in the following time interval of length c. We present an algorithm with the smallest possible competitive ratio for the maximum of delay type objective function. In the case of the sum of delay type objective function we present an $1+O(1/c)$-competitive algorithm. Moreover we show that no algorithm may have smaller competitive ratio than $1+\Omega(1/c^2)$ in the case of that objective function.

Keywords: Online algorithms, lookahead, data acknowledgement.

1 Introduction

In the communication of a computer network the information is sent by packets. If the communication channel is not completely safe then the arrival of the packets must be acknowledged. The TCP implementations are also using acknowledgement of the packets (see [10]). In the data acknowledgement problem we try to determine the time of sending acknowledgements. One acknowledgement can acknowledge many packets but waiting for long time can cause the resending of the packets and that results the congestion of the network. On the other hand sending immediately an acknowledgement about the arrival of each packet would cause again the congestion of the network.

The first online optimization model for determining the sending time of the acknowledgements was developed in [4]. In the model each packet has an arrival time, and at any time the algorithm has to make a decision about the acknowledgement of the arrived packets without any information about the further packets. Two objective functions are investigated. Both of them are the convex combination of the number of acknowledgements and the total latency cost assigned to the acknowledgements (with the coefficients $\gamma, 1-\gamma$). The difference is in the definition of the latency cost assigned to an acknowledgement.

* This research has been supported by the Hungarian National Foundation for Scientific Research, Grant F048587.

L. Kučera and A. Kučera (Eds.): MFCS 2007, LNCS 4708, pp. 288–297, 2007.

In the case of function $f_{\max}$ this is the maximum of the delays which the packets have, in the case of f_{sum} it is the sum of the delays of the packets acknowledged.

Optimal 2-competitive online algorithms are presented in both cases. Semi-online algorithms with some lookahead properties are also considered. In the real application the algorithms have to be online, they do not have lookahead information about the further packets. On the other hand investigating lookahead algorithms is also important for this problem (see [4]). Such algorithms can be used to understand how useful some learning algorithm to estimate the further arrivals of the packets can be. In the case of the $f_{\max}$ objective function it is enough to know the arrival time of the next packet to achieve a 1-competitive algorithm. On the other hand, in the case of the f_{sum} function the knowledge of the next k arrivals is not enough to have better competitive ratio than 2 for any constant k.

In this paper we investigate another version of lookahead property. Instead of giving the knowledge of the arrival times of the next k packets we allow the algorithm to see the arrival times of the packets in a time interval of length c. This type of lookahead is called time lookahead property and it is investigated in [2] for online vehicle routing problems. This new type of lookahead property gives completely different results than the lookahead property investigated in [4]. In the case of the function $f_{\max}$ we can obtain a 1-competitive algorithm if $c \geq \gamma/(1-\gamma)$ and we can define a $2-c(1-\gamma)/\gamma$-competitive algorithm for the smaller values of c. We also prove that smaller competitive ratio can not be achieved. In the case of the function f_{sum} the new lookahead definition allows to achieve smaller competitive ratio than 2. We present an algorithm with a competitive ratio tending 1 in order of magnitude $1 + O(1/c)$ as c is increasing. We also show that no 1-competitive algorithm may exist for constant size of lookahead intervals, we prove a $1 + \Omega(1/c^2)$ lower bound on the possible competitive ratio. We also present the lower bound $2\gamma/(c(1-\gamma)+\gamma)$ for the smaller values of c.

There are some further results on the area of data acknowledgement. The offline version of the problem with the function f_{sum} is further investigated in [8] where a faster, linear time algorithm is developed for its solution. Randomized online algorithms are considered in [7] and [9]. In [7] an $e/(e-1)$-competitive algorithm is given for the solution of the problem. In [9] it is shown that no online randomized algorithm can have smaller competitive ratio than $e/(e-1)$.

Some further objective functions are investigated in [1] and [6]. In [1] the objective function is the sum of the number of acknowledgements and the maximum delay of the packets. A generalized version of the function where the maximum delay is on the power p is also investigated. In both cases optimal online algorithms are presented, in the case of $p = 1$ the algorithm has the competitive ratio $\pi^2/6$, in the general case the competitive ratio can be given by Riemann's zeta function, it tends to 1.5 as p tends to ∞. The paper contains also lower bounds on the competitive ratio of randomized algorithms. In [6] a further cost function is investigated, which can be considered as an extension of $f_{\max}$, and it is also required that the difference between the times of the acknowledgements is at least one time unit. In that paper optimal $(1+\sqrt{5})/2$-competitive deterministic and an optimal $(\sqrt{3}+1)/2$-competitive randomized algorithm are presented. Moreover a class of algorithms which are allowed to use only a limited number of random bits is also investigated. A more general problem than the data acknowledgement problem is investigated in [5]. In that paper an online problem motivated by service call management is considered. The model in the case of two customers can be considered as a generalization of the data acknowledgement problem.

Semi-online algorithms with lookahead are investigated for several online optimization problems. We do not collect all of these results here, we just mention here the most recent paper on online vehicle routing ([2]), one can find further examples in its list of references.

2 Preliminaries

We use the mathematical model which was defined in [4]. In the model the input is the list of the arrival times $a_1, \ldots, a_n$ of the packets. We also denote the packets by their arrival time. The decision maker has to determine when to send acknowledgements, these times are denoted by $t_1, \ldots, t_k$. We consider the objective function

$$\gamma k + (1-\gamma)\sum_{j=1}^{k} L_j,$$

where k is the number of the sent acknowledgements and L_j is the extra latency belonging to the acknowledgement t_j and $\gamma \in (0,1)$ is a constant. We consider two different cases. We obtain the objective function $f_{\max}$ if $L_j = \max_{t_{j-1}<a_i\le t_j}(t_j - a_i)$, the maximal delay collected by j-th acknowledgement. We obtain the objective function f_{sum} if $L_j = \sum_{t_{j-1}<a_i\le t_j}(t_j - a_i)$, the sum of the delays collected by the j-th acknowledgement.

In the on-line problem at time t the decision maker only knows the arrival times of the packets already arrived and has no information about the further packets. We consider a semi-online model with time lookahead c, where at time t the decision maker knows the arrival times of the packets already arrived and also knows the arrival times of the packets arriving in the time interval $(t, t+c]$. We denote the set of the unacknowledged packets at the arrival time a_i by σ_i. For an arbitrary list L of packets and an algorithm A, we denote by $A(L)$ the total cost of the acknowledgements sent by algorithm A on list L. The total cost of sending optimally the acknowledgements is denoted by $OPT(L)$. We use the competitive analysis to evaluate the algorithms, as it is usual in the case of online and semi-online algorithms ([3]). An algorithm is d-competitive if $A(I) \le d \cdot OPT(I)$ is valid for every input I. The competitive ratio of an algorithm is the smallest number d for which the algorithm is d-competitive.

We will examine time lookahead extensions of the online alarming algorithms defined in [4], thus we recall here the definition of these algorithms. An alarming algorithm works as follows. At the arrival time a_j an alarm is set for time $a_j + e_j$. If no packet arrives before time $a_j + e_j$, then an acknowledgement is sent at time $a_j + e_j$ which acknowledges all of the unacknowledged packets. Otherwise at the arrival of the next packet at time a_{j+1} the alarm is reset for time $a_{j+1} + e_{j+1}$.

These algorithms are defined and analysed in [4]. In the case of function $f_{\max}$ the alarming algorithm which uses the value $e_j = \gamma/(1-\gamma)$ for each j is an optimal 2-competitive algorithm. In the case of function f_{sum} the alarming algorithm which uses the value $e_j = (\gamma/(1-\gamma) - \sum_{a_i \in \sigma_j}(a_j - a_i))/|\sigma_j|$ for each j is an optimal 2-competitive algorithm. (It is worth noting that in this case e_j is chosen to balance the two types of cost, if no further packet arrives than sending an acknowledgement at time $a_j + e_j$ has latency cost γ).

3 The $f_{\max}$ Objective Function

3.1 Algorithm

In [4] it is shown that the optimal offline solution has a very simple structure in the case of function $f_{\max}$. The following statement is valid.

Proposition 1. *Under $f_{\max}$ there exists an optimal solution S that places an acknowledgement at a_j if and only if $a_{j+1} - a_j \geq \gamma/(1-\gamma)$.*

We consider two cases depending on the value of c. When $c \geq \gamma/(1-\gamma)$, then we obtain a 1-competitive algorithm easily. The size of the lookahead interval is large enough to determine whether $a_{j+1} - a_j > \gamma/(1-\gamma)$ is valid or not, thus an algorithm with lookahead c can find the optimal solution described in Proposition 1.

The case when $c < \gamma/(1-\gamma)$ is more interesting. We define for this case an extended version of the alarming algorithm developed in [4]. This *time lookahead alarming algorithm* (TLA in short) works as follows. At the arrival time a_j an alarm is set for time $a_j + \gamma/(1-\gamma) - c$. If the packet a_{j+1} arrives before the alarm or we can see a_{j+1} at time $a_j + \gamma/(1-\gamma) - c$ in the lookahead interval $(a_j + \gamma/(1-\gamma) - c, a_j + \gamma/(1-\gamma)]$ then we postpone the alarm to the time $a_{j+1} + \gamma/(1-\gamma) - c$. In the opposite case (no packet arrives in the time interval $(a_j, a_j + \gamma/(1-\gamma)]$) an acknowledgement is sent at time $a_j + \gamma/(1-\gamma) - c$ which acknowledges all of the unacknowledged packets.

Theorem 1. *TLA is* $\max\{1, 2 - \frac{1-\gamma}{\gamma}c\}$*-competitive.*

Proof. First we show that the algorithm is $2 - \frac{1-\gamma}{\gamma}c$-competitive. Consider an arbitrary input sequence $a_1, \ldots, a_n$. Partition the sequence into phases. Let $S_1 = \{a_1, \ldots, a_{k(1)}\}$ where $k(1)$ is the first index with the property $a_{k(1)+1} - a_{k(1)} \geq \gamma/(1-\gamma)$. The other phases are defined in the same way $S_{j+1} = \{a_{k(j)+1}, \ldots, a_{k(j+1)}\}$ where $k(j+1)$ is the first index after $k(j)$ with the property $a_{k(j+1)+1} - a_{k(j+1)} \geq \gamma/(1-\gamma)$. The last phase is ended by the last packet. We will also use the value $k(0) = 0$.

Then an optimal offline algorithm sends an acknowledgement at the last packet of each phase. Therefore it has the total cost $OPT(S_j) = \gamma + (1-\gamma)(a_{k(j)} - a_{k(j-1)+1})$ for the acknowledgement of the j-th phase. On the other hand TLA sends the acknowledgement for the phase at time $a_{k(j)} + \gamma/(1-\gamma) - c$, thus it has a total cost $TLA(S_j) = \gamma + (1-\gamma)(a_{k(j)} + \gamma/(1-\gamma) - c - a_{k(j-1)+1})$ for the acknowledgement of the j-th phase. We have $TLA(S_j)/OPT(S_j) > 1$, thus decreasing both values by the same constant increases the ratio. Therefore we obtain that

$$\frac{TLA(S_j)}{OPT(S_j)} \leq \frac{\gamma + (1-\gamma)(\gamma/(1-\gamma) - c)}{\gamma} = 2 - \frac{1-\gamma}{\gamma}c.$$

Since the total cost is the sum of the costs of the phases, thus we obtain that TLA is $2 - \frac{1-\gamma}{\gamma}c$-competitive.

3.2 Lower Bound

TLA has the smallest possible competitive ratio, as the following statement shows.

Theorem 2. *No semi-online algorithm with lookahead c may have smaller competitive ratio than* $\max\{1, 2 - \frac{1-\gamma}{\gamma}c\}$.

Proof. If $c \geq \gamma/(1-\gamma)$ then the statement is obviously true, thus we can assume that $c < \gamma/(1-\gamma)$. Consider an arbitrary online algorithm, denote it by A. Define the following input sequence denoted by I_n. The arrival time of the first packet is $a_1 = 0$, then the i-th packet arrives ($i = 2, \ldots, n$) c time units after the acknowledgement of the $i-1$-th packet ($a_i = t_{i-1} + c$).

We partition the input sequence into phases in the same way as in the proof of Theorem 1. Denote the phases by $S_1, \ldots, S_j$. Consider the phase S_i. To simplify the notation denote the value $k(i) - k(i-1)$ by r_i. Algorithm A sends an acknowledgement for each packet thus we obtain that

$$A(S_i) = \gamma r_i + (1-\gamma) \sum_{p=k(i-1)+1}^{k(i)} (t_p - a_p).$$

The optimal solution sends only one acknowledgement at time $a_{k(i)}$ thus

$$OPT(S_i) = \gamma + (1-\gamma)\big(\sum_{p=k(i-1)+1}^{k(i)-1} (t_p - a_p) + c(r_i - 1)\big).$$

Now suppose that $i < j$. If we calculate $A(S_i) - (2 - \frac{1-\gamma}{\gamma}c)OPT(S_i)$ and we use that $t_p - a_p \leq \gamma/(1-\gamma) - c$ for $p = k(i-1)+1, \ldots, k(i)-1$ and $t_{k(i)} - a_{k(i)} > \gamma/(1-\gamma) - c$ then we obtain that

$$A(S_i) - (2 - \frac{1-\gamma}{\gamma}c)OPT(S_i) \geq \gamma(r_i - 2 + \frac{1-\gamma}{\gamma}c) + (1-\gamma)\Bigg(\big(\frac{1-\gamma}{\gamma}c - 2\big)(r_i - 1)c +$$

$$(r_i - 1)\big(\frac{1-\gamma}{\gamma}c - 1\big)\big(\frac{\gamma}{1-\gamma} - c\big) + \frac{\gamma}{1-\gamma} - c\Bigg) = \gamma(r_i - 2 + \frac{1-\gamma}{\gamma}c) -$$

$$(1-\gamma)(r_i - 1)c + (r_i - 1)((1-\gamma)c - \gamma) + \gamma - c(1-\gamma) = 0.$$

In the second equality we simplified the formula by eliminating $(r_i-1)(\frac{1-\gamma}{\gamma}c-1)c$ which can be found in the formula with the coefficients +1 and −1.

Therefore we proved that $A(S_i) \geq (2-\frac{1-\gamma}{\gamma}c)OPT(S_i)$ if $i < j$. In the case of S_j there is only one difference, we cannot state that the inequality $t_{k(j)} - a_{k(j)} > \gamma/(1-\gamma) - c$ is valid. But we can prove in the same way as above that $A(S_j) + \gamma/(1-\gamma) - c \geq (2-\frac{1-\gamma}{\gamma}c)OPT(S_j)$.

Since the total cost is the sum of the costs of the phases, thus we obtain that $A(I_n) + \gamma/(1-\gamma) - c \geq (2-\frac{1-\gamma}{\gamma}c)OPT(I_n)$. On the other hand as n tends to ∞ the value of $OPT(I_n)$ also tends to ∞, thus the above inequality shows that $A(I_n)/OPT(I_n)$ can be arbitrarily close to $2-\frac{1-\gamma}{\gamma}c$ thus the competitive ratio of A can not be smaller than $2-\frac{1-\gamma}{\gamma}c$.

4 The Sum Objective Function

4.1 Algorithms

It is a straightforward idea to also use the time lookahead extension of the alarming algorithm from [4] in this case. We can define the TLA extension of the alarming algorithm for this case as follows. At the arrival time a_j an alarm is set for time $a_j + e_j$ where $e_j = (\gamma/(1-\gamma) - \sum_{a_i \in \sigma_j}(a_j - a_i))/|\sigma_j|$. If the packet a_{j+1} arrives before the time $\max\{a_j, a_j+e_j-c\}$ or we can see a_{j+1} at this time in the lookahead interval, then we move to a_{j+1} and reset the alarm. In the opposite case (no packet arrives in the time interval $(a_j, a_j+e_j])$ an acknowledgement is sent at time $\max\{a_j, a_j+e_j-c\}$ which acknowledges all of the unacknowledged packets. Unfortunately this lookahead extension of the algorithm does not make it possible to achieve a smaller competitive ratio than 2.

Theorem 3. *The competitive ratio of TLA is 2 for arbitrary c.*

Proof. The cost of this algorithm is at most the cost of the online alarming algorithm from [4] on any input, thus it follows immediately that TLA is 2-competitive from the result that the online alarming algorithm is 2-competitive. TLA has no better performance as the following input sequence shows. Let $I_n = \{a_1, \ldots, a_{2n+1}\}$ where $a_1 = 0$ and $a_{2i} = i\gamma/(1-\gamma) + (i-1)\varepsilon$, $a_{2i+1} = i\gamma/(1-\gamma) + i\varepsilon$ for $i = 1, \ldots, n$. Then TLA sends the acknowledgements at $a_2, \ldots, a_{2n}, \max\{a_{2n+1}, a_{2n+1} + \gamma/(1-\gamma) - c\}$. Thus $TLA(I_n) = (n+1)\gamma + n\gamma + (1-\gamma)\max\{0, \gamma/(1-\gamma) - c\}$. An optimal offline algorithm sends an acknowledgement at a_1, a_3, a_{2n+1} and it has the cost $OPT(I_n) = (n+1)\gamma + (1-\gamma)n\varepsilon$. The ratio of the costs tends to 2 as ε tends to 0 and n tends to ∞, and this yields that the competitive ratio is at least 2.

In the case when $c > \gamma/(1-\gamma)$ we can achieve smaller competitive ratio than 2 by the following algorithm. We present the Lookahead Interval Planning Algorithm (LIP in short). The algorithm partitions the packets into blocks and for each block determines the acknowledgments based on an offline optimal solution. The block always starts at the first unacknowledged packet. First the algorithm examines whether there exist packets a_i, a_{i+1} in the c length lookahead interval with the property $a_{i+1}-a_i > \gamma/(1-\gamma)$. If there exists such pair, then the block is ended at the first such a_i, otherwise the block has length c. Then LIP calculates the optimal solution of the offline acknowledgement problem for the packets in the block, it can use one of the algorithms which solves the offline problem (such algorithms are presented in [4] and [8]) and sends the acknowledgements according to this solution and considers the next block.

Theorem 4. *LIP is* $1+\frac{\gamma}{(1-\gamma)c}$*-competitive.*

Proof. To prove that LIP is $1+\frac{\gamma}{(1-\gamma)c}$-competitive consider an arbitrary input I. Divide the input into phases in the same way as in the proof of Theorem 1. Let us observe that there exists an offline optimal algorithm which sends an acknowledgement at the last packet of each phase. (Because, if last packet of the phase is delayed, it incurs a delay cost of more than $(1-\gamma)(\gamma/(1-\gamma)) = \gamma$, whereas it incurs a communication cost of exactly γ if it is acknowledged. Furthermore let us observe that the last packet of a phase is always a last packet of some block, thus LIP also sends an acknowledgement at the last packet of the phase.

Consider an arbitrary phase S_i. Denote by r the number of blocks in the phase. Consider an optimal solution of the phase. If we extend it with $r-1$ further acknowledgements on the end of the first $r-1$ blocks, then we obtain a solution which acknowledges the blocks separately. But LIP gives the best such solution therefore we obtain that $(r-1)\gamma + OPT(S_i) \geq LIP(S_i)$ which yields that $LIP(S_i)/OPT(S_i) \leq 1 + (r-1)\gamma/OPT(S_i)$.

Consider the value of $OPT(S_i)$. Since each block is in the same phase, thus the length of each of the first $r-1$ blocks is c, therefore the length of the phase is at least $(r-1)c$. Suppose that the optimal offline algorithm sends k acknowledgements in this phase. Then after each of the first $k-1$ acknowledgement there is an at most $\gamma/(1-\gamma)$ length interval without packet. This yields that in this case the total latency cost of OPT is at least $(1-\gamma)((r-1)c-(k-1)\gamma/(1-\gamma))$. Therefore $OPT(S_i) \geq k\gamma + (1-\gamma)((r-1)c-(k-1)\gamma/(1-\gamma)) = (1-\gamma)(r-1)c+\gamma$. On the other hand if we use this bound we obtain that $LIP(S_i)/OPT(S_i) \leq 1+\frac{\gamma}{(1-\gamma)c}$.

4.2 Lower Bounds

First we give a lower bound on the order of magnitude of the possible competitive ratio. This bound is useful for large lookahead intervals, it shows that no constant size lookahead is enough to achieve 1-competitiveness in the case of the f_{sum} function.

Theorem 5. *No online algorithm may have smaller competitive ratio than* $1+\Omega(1/c^2)$.

Proof. To simplify the calculation suppose that $c = \gamma(k-1/4)/(1-\gamma)$, where $k \geq 1$ is an integer. We can assume that without loss of generality since allowing larger lookahead can not increase the competitive ratio and we are proving lower bound on the order of magnitude of the possible competitive ratio. Consider an arbitrary algorithm A with lookahead c. Let $x = \frac{\gamma(2k+1)}{4k(1-\gamma)}$ and $y = \gamma/2(1-\gamma)$. Define the following sequences for each $j = 1, \ldots, k$.

- $S_{xjx} = \{a_1, a_2, \ldots, a_{2j}\}$ where $a_{2i-1} = (i-1)x+(i-1)y$ and $a_{2i} = ix+(i-1)y$ for $i = 1, \ldots, j$. Let us note that $a_{2k} = c$ in S_{xkx}.
- $S_{xjy} = \{a_1, a_2, \ldots, a_{2j+1}\}$ where $a_{2i-1} = (i-1)x+(i-1)y$ for $i = 1, \ldots, j+1$ and $a_{2i} = ix + (i-1)y$ for $i = 1, \ldots, j$.
- $S_{yjy} = \{a_1, a_2, \ldots, a_{2j}\}$ where $a_{2i-1} = (i-1)y+(i-1)x$ and $a_{2i} = iy+(i-1)x$ for $i = 1, \ldots, j$.
- $S_{yjx} = \{a_1, a_2, \ldots, a_{2j+1}\}$ where $a_{2i-1} = (i-1)y+(i-1)x$ for $i = 1, \ldots, j+1$ and $a_{2i} = iy + (i-1)x$ for $i = 1, \ldots, j$.

Since $\gamma + (1-\gamma)(2y + x) = 2\gamma + (1-\gamma)x$, thus we obtain that there exist optimal solutions for the above defined sequences which never acknowledge more than 2 packets with one acknowledgement. Using this observation the following lemma can be easily proven by induction.

Lemma 1. *For each* $1 \leq j \leq k$ *we have* $OPT(S_{xjy}) = OPT(S_{yjx}) = \gamma(j+1) + (1-\gamma)jy$, $OPT(S_{xjx}) = \gamma j + (1-\gamma)jx$, $OPT(S_{yjy}) = \gamma j + (1-\gamma)jy$.

Give S_{xkx} as the first part of the input to A and wait till time y. Suppose that the algorithm sends an acknowledgement at time $z \leq y$. Then it acknowledges the packet a_1 and it has to handle the remaining part which is $S_{y(k-1)x}$. Therefore by Lemma 1 we obtain that $A(S_{xkx}) \geq \gamma + (1-\gamma)z + k\gamma + (1-\gamma)(k-1)y$. Therefore we obtain that

$$\frac{A(S_{xkx})}{OPT(S_{xkx})} \geq \frac{(k+1)\gamma + (1-\gamma)(k-1)y}{k\gamma + k(1-\gamma)x} = 1 + \frac{1}{6k+1}.$$

Now suppose that A does not send an acknowledgement before time y. Then at time $y + c$ a further packet arrives, thus the input is S_{xky}. The algorithm observes this packet at time y. If it acknowledges the first packet before time x then $A(S_{xky}) \geq \gamma + (1-\gamma)y + OPT(S_{yky})$. Therefore by Lemma 1 we obtain that the following inequality for this case:

$$\frac{A(S_{xky})}{OPT(S_{xky})} \geq \frac{(k+1)\gamma + (1-\gamma)(k+1)y}{(k+1)\gamma + (1-\gamma)ky} = 1 + \frac{1}{3k+2}$$

Finally, suppose that A does not send an acknowledgement before time x. If it sends its first acknowledgement later than x then its total cost is increasing, therefore we can assume that the first acknowledgement is sent at time x. Then the algorithm acknowledges the first two packets and the remaining part is $S_{x(k-1)y}$, thus we obtain $A(S_{xky}) \geq \gamma + (1-\gamma)x + OPT(S_{x(k-1)y})$. Therefore by Lemma 1 we obtain that the following inequality for this case:

$$\frac{A(S_{xky})}{OPT(S_{xky})} \geq \frac{(k+1)\gamma + (1-\gamma)(x + (k-1)y)}{(k+1)\gamma + (1-\gamma)ky} \geq 1 + \frac{1}{6k^2 + 4k}$$

Since we examined all of the possible cases, we proved the statement of the theorem.

The above bound does not give better value than 1 in the case when the algorithm has only a small lookahead interval. We also show a lower bound for $c \leq \gamma/(1-\gamma)$ by extending the technique which is used to prove a lower bound in [4].

Theorem 6. *No online algorithm with lookahead c can have smaller competitive ratio than $2\gamma/(c(1-\gamma)+\gamma)$.*

Proof. Consider an arbitrary on-line algorithm denote it by A. Analyze the following input. Consider a long sequence of packets where the packets always arrive c time units after the time when A sends an acknowledgement $(t_j + c = a_{j+1})$. Then the on-line cost of a sequence containing $2n+1$ packets is $A(I_{2n+1}) = \gamma(2n+1) + (1-\gamma)\sum_{i=1}^{2n+1}(t_i - a_i)$. Consider the following two offline algorithms. ODD sends the acknowledgements after the odd numbered packets and after the last packet, and EV sends the acknowledgements after the even numbered packets.

Then the costs achieved by these algorithms are

$$\mathrm{EV}(I_{2n+1}) = (n+1)\gamma + (1-\gamma)\sum_{i=1}^{n}(a_{2i+1} - a_{2i}) = (n+1)\gamma + (1-\gamma)(nc + \sum_{i=1}^{n}(t_{2i} - a_{2i}))$$

and

$$\mathrm{ODD} = (n+1)\gamma + (1-\gamma)\sum_{i=1}^{n}(a_{2i} - a_{2i-1}) = (n+1)\gamma + (1-\gamma)(nc + \sum_{i=1}^{n}(t_{2i-1} - a_{2i-1})).$$

On the other hand none of the costs achieved by ODD and EV is smaller than the optimal offline cost, thus $OPT(I_{2n+1}) \leq \min\{EV(I_{2n+1}), \mathrm{ODD}(I_{2n+1})\} \leq (EV(I_{2n+1}) + ODD(I_{2n+1}))/2$. Therefore we obtain that

$$\frac{A(I_{2n+1})}{OPT(I_{2n+1})} \geq \frac{2(\gamma(2n+1) + (1-\gamma)\sum_{i=1}^{2n+1}(t_i - a_i))}{\gamma(2n+2) + (1-\gamma)(\sum_{i=1}^{2n}(t_i - a_i) + 2nc)} \geq 2 - \frac{2\gamma + 4nc(1-\gamma)}{\gamma(2n+2) + 2nc(1-\gamma)}.$$

The ratio which we obtained as a lower bound on $A(I_{2n+1})/OPT(I_{2n+1})$ tends to $2\gamma/(c(1-\gamma)+\gamma)$ as n tends to ∞, and this proves the theorem.

References

1. Albers, S., Bals, H.: Dynamic TCP acknowledgement: Penalizing long delays. SIAM J. Discrete Math. 19(4), 938–951 (2005)
2. Allulli, L., Ausiello, G., Laura, L.: On the power of lookahead in on-line vehicle routing problems. In: Wang, L. (ed.) COCOON 2005. LNCS, vol. 3595, pp. 728–736. Springer, Heidelberg (2005)
3. Borodin, A., El-Yaniv, R.: Online Computation and Competitive Analysis, Cambridge University Press (1998)
4. Dooly, D.R., Goldman, S.A., Scott, S.D.: On-line analysis of the TCP acknowledgment delay problem. J. ACM 48(2), 243–273 (2001)
5. Epstein, L., Kesselman, A.: On the remote server problem or more about TCP acknowledgments. Theoretical Computer Science 369(1-3), 285–299 (2006)

6. Frederiksen, J.S., Larsen, K.S., Noga, J., Uthaisombut, P.: Dynamic TCP acknowledgment in the LogP model. Journal of Algorithms 48(2), 407–428 (2003)
7. Karlin, A.R., Kenyon, C., Randall, D.: Dynamic TCP acknowledgement and other stories about e/(e-1). Algorithmica 36, 209–224 (2003)
8. Noga, J., Seiden, S.S., Woeginger, G.J.: A faster off-line algorithm for the TCP acknowledgement problem. Information Processing Letters 81(2), 71–73 (2002)
9. Seiden, S.S.: A guessing game and randomized online algorithms. In: Proceedings of STOC, pp. 592–601 (2000)
10. Stevens, W.R.: TCP/IP Illustrated, Volume I. The protocols, Addison Wesley, Reading (1994)