Dynamic TCP Acknowledgement: Penalizing Long Delays

Karousatou Christina

Network Algorithms

June 8, 2010

Layout

- Data transmission in networks using Transmission Control Protocol (TCP).
- The data is partitioned into segments or *packets* that are sent across the connection.
- Each received packet must be acknowledged so that the sending node is notified that the transmission was successful.

- Acknowledging each packet individually has enough disadvantages, e.g.
  - Network congestion
  - Overhead at the network nodes for sending and receiving acknowledgements
- The idea is to reduce the number of acknowledgements by employing some delay mechanism so that the TCP can acknowledge multiple incoming packets with a single acknowledgement.
- This mechanism, though, adds latency to a TCP connection.
- The goal is to balance the reduction in the number of acknowledgements with the increase in the latency.

Dooly, Goldman and Scott formulated the following problem.

- A network node receives a sequence of n data packets.
- Let α_i denote the arrival time of packet i, $1 \leq i \leq n$.
- At time α_i, the arrival times α_j, $j > i$, are not known.
- We have to partition the sequence $\sigma = (\alpha_1, \ldots, \alpha_n)$ into m subsequences $\sigma_1, \ldots, \sigma_m$, $m \geq 1$, such that each subsequence ends with an acknowledgement.
- Finally, let t_i be the time when the acknowledgement for σ_i is sent and we require $t_i \geq \alpha_j$, for all $\alpha_j \in \sigma_i$.

Layout

Definition

Given a solution generated by an acknowledgement algorithm $\mathcal{A}$ on input σ, the resulting objective function value is also referred to as the $cost$ $\mathcal{C}_{\mathcal{A}}(\sigma)$ of $\mathcal{A}$ on σ.
An online algorithm $\mathcal{A}$ is called $c - competitive$ if there exists a constant b such that $\mathcal{C}_{\mathcal{A}}(\sigma) \leq c \cdot \mathcal{C}_{\mathcal{OPT}}(\sigma) + b$, for all inputs σ.
Here $\mathcal{C}_{\mathcal{OPT}}(\sigma)$ is the cost incurred by an optimal offline algorithm that knows the entire input σ in advance and can serve it with minimum cost.

- Previous work has focused mostly on the objective function that minimizes the number of acknowledgements and the sum of the delays incurred for all of the packets, i.e.
  - $h = m + \sum_{i=1}^{m} \sum_{a_j \in \sigma_i} (t_i - a_j)$
  - $h' = m + \sum_{i=1}^{m} \max_{a_j \in \sigma_i} (t_i - a_j)$

- Previous work has focused mostly on the objective function that minimizes the number of acknowledgements and the sum of the delays incurred for all of the packets, i.e.
  - $h = m + \sum_{i=1}^{m} \sum_{a_j \in \sigma_i} (t_i - a_j)$
  - $h' = m + \sum_{i=1}^{m} \max_{a_j \in \sigma_i} (t_i - a_j)$
- Dooly et al. presented a deterministic 2-competitive online algorithm for the objective function h (with or without bounded lookahead) and showed that no deterministic online strategy can achieve a smaller competitive ratio.
  Also, studied the minimization of the objective function h' and showed that the best competitive ratio of a deterministic algorithm without lookahead is equal to 2.

- Most implementations of TCP have a $maximum\ delay\ constraint$.

- Most implementations of TCP have a $maximum\ delay\ constraint$.
- In this case, Dooly et al. showed that their algorithm can be modified and remains 2-competitive.

- Most implementations of TCP have a $maximum\ delay\ constraint$.
- In this case, Dooly et al. showed that their algorithm can be modified and remains 2-competitive.
- Karlin, Kenyon and Randall developed a randomized online strategy that achieves $e/(e-1)$-competitiveness (≈ 1.58). Also, pointed out that the TCP acknowledgement problem with objective functions h and h' are ski rental type problems.
  Noga and independently Seiden showed that no randomized algorithm can do better.

Layout

- In this paper Albers and Bals study the objective functions

$$f = m + \max_{1 \leq i \leq m} d_i$$

With $d_i = \max_{a_j \in \sigma_i}(t_i - a_j)$, $1 \leq i \leq m$.
And

$$f_p = m + \max_{1 \leq i \leq m} {d_i}^p$$

Where delays in this function are penalized more heavily.

- In this paper Albers and Bals study the objective functions

$$f = m + \max_{1 \leq i \leq m} d_i$$

With $d_i = \max_{a_j \in \sigma_i}(t_i - a_j)$, $1 \leq i \leq m$.
And

$$f_p = m + \max_{1 \leq i \leq m} {d_i}^p$$

Where delays in this function are penalized more heavily.

- Note that the current problem is not a ski rental problem.

Layout

Let z be a positive real number.

Algorithm Linear-Delay(z)

Initially, set $d = z$ and send the first acknowledgement at time $a_1 + d$. In general, suppose that the i-th acknowledgement has just been sent and that j packets have been processed so far. Set $d = (i + 1)z$ and send the $(i + 1)$-st acknowledgement at time $a_{j+1} + d$.

Theorem 2.1.

For any z with $z \geq 1/2$, Linear-Delay(z) is c-competitive, where $c = \max\{1 + z, (1 + z)/(2 + z - \pi^2/6)\}$.

Corollary 2.1.

Setting $z = \pi^2/6 - 1$, Linear-Delay(z) achieves a competitive ratio of $\pi^2/6 \approx 1.644$.

Proof

In the following we call the online algorithm **LD**(z) for short.

- Suppose that **LD**(z) serves the input sequence using m acknowledgements.
  The online cost is $\mathcal{C}_{LD(z)}(\sigma) = m(1+z)$.

Proof

In the following we call the online algorithm **LD**(z) for short.

- Suppose that **LD**(z) serves the input sequence using m acknowledgements.
  The online cost is $\mathcal{C}_{LD(z)}(\sigma) = m(1+z)$.
- We have to lower bound the cost incurred by an optimal offline algorithm $\mathcal{OPT}$.
  Suppose that the optimum offline algorithm uses l acknowledgements and that the maximum acknowledgement delay is $\mathcal{C}$, $\mathcal{C} \geq 0$. Then $\mathcal{C}_{\mathcal{OPT}}(\sigma) = l + \mathcal{C}$.

Proof

Definition

In the sequence of n packets we identify a subsequence of m $main$ $packets$, numbered from 0 to $m-1$. Main packet 0 is the first packet in the input sequence. Main packet i, $1 \leq i \leq m-1$, is the first packet that arrives after the i-th acknowledgement sent by **LD**(z).

Note that the time difference between the $(i-1)$-st and the i-th main packets is larger than iz, for $i = 1, \ldots, m-1$.

Proof

Definition

Associated with each acknowledgement α sent by $\mathcal{OPT}$ is an *acknowledgement interval* that starts when the first packet acknowledged by α arrives and ends when α is sent. The length of each interval is bounded by $\mathcal{C}$.

Lemma 2.1.

Any acknowledgement interval starting at or after the arrival of main packet $\lfloor \frac{\mathcal{C}}{iz} \rfloor$ can contain at most i main packets.

Proof

Proof of Lemma 2.1.

Main packet k with $k \geq \lfloor \frac{\mathcal{C}}{iz} \rfloor + 1$ has a distance of more than $z(\lfloor \frac{\mathcal{C}}{iz} \rfloor + 1)$ to the previous main packet. If the ackowledgement interval contained at least $i+1$ main packets, then the length of the interval would be at least $iz(\lfloor \frac{\mathcal{C}}{iz} \rfloor + 1) > iz(\frac{\mathcal{C}}{iz}) = \mathcal{C}$, which is impossible. $\square$

Proof

Proof of Lemma 2.1.

Main packet k with $k \geq \lfloor \frac{\mathcal{C}}{iz} \rfloor + 1$ has a distance of more than $z(\lfloor \frac{\mathcal{C}}{iz} \rfloor + 1)$ to the previous main packet. If the ackowledgement interval contained at least $i+1$ main packets, then the length of the interval would be at least $iz(\lfloor \frac{\mathcal{C}}{iz} \rfloor + 1) > iz(\frac{\mathcal{C}}{iz}) = \mathcal{C}$, which is impossible. $\square$

Define $i_0 = \lfloor \sqrt[3]{\frac{\mathcal{C}}{z}} \rfloor - 1$. In the rest of this proof we assume $i_0 \geq 2$. If $i_0 \leq 1$, then $\mathcal{C} \leq 27z$ and $\mathcal{OPT}$ must acknowledge each of the last $m - 27z$ main packets with seperate acknowledgements. In this case **LD**(z) is clearly $(1+z)$-competitive.

Proof

Lemma 2.2.

Let $1 \leq i \leq i_0$. The acknowledgement interval containing main packet k, for $k \geq \lfloor \frac{\mathcal{C}}{iz} \rfloor$, must have started after the arrival of main packet $\lfloor \frac{\mathcal{C}}{(i+1)z} \rfloor$.

Proof

Lemma 2.2.

Let $1 \leq i \leq i_0$. The acknowledgement interval containing main packet k, for $k \geq \lfloor \frac{\mathcal{C}}{iz} \rfloor$, must have started after the arrival of main packet $\lfloor \frac{\mathcal{C}}{(i+1)z} \rfloor$.

Proof of Lemma 2.2.

If not, the nunber of main packets in this time window would be $\lfloor \frac{\mathcal{C}}{iz} \rfloor - \lfloor \frac{\mathcal{C}}{(i+1)z} \rfloor + 1 > \lfloor \frac{\mathcal{C}}{i(i+1)z} \rfloor \geq i + 2$. The last inequality is equivalent to $\mathcal{C}/z \geq i(i+1)(i+2)$ and holds for all $i \leq i_0$. Thus there are at least $i + 2$ main packets in this time window. Each of the last $i + 1$ of these is more than $z(\lfloor \frac{\mathcal{C}}{(i+1)z} \rfloor + 1)$ time units away from the previous main packet and thus the length of the window is greater than $(i+1)z(\lfloor \frac{\mathcal{C}}{(i+1)z} \rfloor + 1) > (i+1)z(\frac{\mathcal{C}}{(i+1)z}) = \mathcal{C}$, which is impossible. □

Proof

In order to estimate the number of acknowledgements sent by $\mathcal{OPT}$, we use the following charging scheme.

Proof

In order to estimate the number of acknowledgements sent by $\mathcal{OPT}$, we use the following charging scheme.

- Consider an ackowledgement costs 1.

Proof

In order to estimate the number of acknowledgements sent by $\mathcal{OPT}$, we use the following charging scheme.

- Consider an ackowledgement costs 1.
- We charge this cost to the main packets contained in the associated acknowledgement interval and split the cost evenly among these main packets.

Proof

In order to estimate the number of acknowledgements sent by $\mathcal{OPT}$, we use the following charging scheme.

- Consider an ackowledgement costs 1.
- We charge this cost to the main packets contained in the associated acknowledgement interval and split the cost evenly among these main packets.
- If an acknowledgement interval does not contain a main packet, then we ignore it in the analysis of $\mathcal{OPT}$'s cost.

Proof

In order to estimate the number of acknowledgements sent by $\mathcal{OPT}$, we use the following charging scheme.

- Consider an ackowledgement costs 1.
- We charge this cost to the main packets contained in the associated acknowledgement interval and split the cost evenly among these main packets.
- If an acknowledgement interval does not contain a main packet, then we ignore it in the analysis of $\mathcal{OPT}$'s cost.

Summing over all main packets, we derive a lower bound on the optimum cost incurred for sending acknowledgements.

Proof

We assume that $\mathcal{C} < m$. If $\mathcal{C} \geq m$, then **LD**(z) is clearly $(1+z)$-competitive because **LD**(z)'s cost is $(1+z)m$ and the optimum offline cost is at least m.

Proof

We assume that $\mathcal{C} < m$. If $\mathcal{C} \geq m$, then **LD**(z) is clearly $(1+z)$-competitive because **LD**(z)'s cost is $(1+z)m$ and the optimum offline cost is at least m.

First case : $\mathcal{C} \leq zm$

Each main packet is contained in some acknowledgement interval. Let i be an integer with $1 \leq i \leq i_0$. We analyze the cost charged to main packet k with $k \geq \lfloor \frac{\mathcal{C}}{iz} \rfloor$ and $k < \lfloor \frac{\mathcal{C}}{(i-1)z} \rfloor$. If $i = 1$, then $k < m$.

Proof

- If the acknowledgement interval containing main packet k started at or after the arrival of main packet $\lfloor \frac{\mathcal{C}}{iz} \rfloor$, then by Lemma 2.1 at most i main packets are contained in the interval and main packet k is assigned a cost of at least $1/i$.

Proof

- If the acknowledgement interval containing main packet k started at or after the arrival of main packet $\lfloor \frac{\mathcal{C}}{iz} \rfloor$, then by Lemma 2.1 at most i main packets are contained in the interval and main packet k is assigned a cost of at least $1/i$.

- If the ackowledgement interval started earlier, then by Lemma 2.2 it must have started after the arrival of main packet $\lfloor \frac{\mathcal{C}}{(i+1)z} \rfloor$. Applying Lemma 2.1 for $i+1$, we obtain that the interval contains at most $i+1$ main packets and the packet is assigned a cost of at least $1/(i+1)$.

Proof

There is only one acknowledgement interval that starts before and ends after the arrival of main packet $\lfloor \frac{\mathcal{C}}{iz} \rfloor$. Thus for at most $i+1$ main packets considered above, the cost is lower bounded by $1/(i+1)$ instead of $1/i$.

Proof

There is only one acknowledgement interval that starts before and ends after the arrival of main packet $\lfloor \frac{\mathcal{C}}{iz} \rfloor$. Thus for at most $i+1$ main packets considered above, the cost is lower bounded by $1/(i+1)$ instead of $1/i$.

For $i = 1$, the total cost assigned to all main packets k with $k \geq \lfloor \frac{\mathcal{C}}{z} \rfloor$ is

$$(m - \lfloor \frac{\mathcal{C}}{z} \rfloor) - 2(1 - \frac{1}{2}) = (m - \lfloor \frac{\mathcal{C}}{z} \rfloor) - 1$$

Proof

For $2 \leq i \leq i_0$, the total cost assigned to main packet k, $\lfloor \frac{\mathcal{C}}{iz} \rfloor \leq k < \lfloor \frac{\mathcal{C}}{(i-1)z} \rfloor$ is at least

$$(\lfloor \frac{\mathcal{C}}{(i-1)z} \rfloor - \lfloor \frac{\mathcal{C}}{iz} \rfloor)\frac{1}{i} - (i+1)(\frac{1}{i} - \frac{1}{i+1})$$

$$= (\lfloor \frac{\mathcal{C}}{(i-1)z} \rfloor - \lfloor \frac{\mathcal{C}}{iz} \rfloor)\frac{1}{i} - \frac{1}{i}$$

Proof

Summing over all i, we obtain that the number of acknowledgements sent by $\mathcal{OPT}$ is at least

$$l \geq m - \lfloor \frac{\mathcal{C}}{z} \rfloor - 1 + \sum_{i=2}^{i_0} \left((\lfloor \frac{\mathcal{C}}{(i-1)z} \rfloor - \lfloor \frac{\mathcal{C}}{iz} \rfloor) \frac{1}{i} - \frac{1}{i} \right)$$

$$= m - \sum_{i=1}^{i_0-1} \lfloor \frac{\mathcal{C}}{iz} \rfloor \left(\frac{1}{i} - \frac{1}{i+1} \right) - \lfloor \frac{\mathcal{C}}{i_0 z} \rfloor \frac{1}{i_0} - H_{i_0}$$

Here H_{i_0} denotes the i_0-th Harmonic number. Thus,

Proof

$$l \geq m - \frac{\mathcal{C}}{z}\sum_{i=1}^{\infty}\left(\frac{1}{i^2} - \frac{1}{i(i+1)}\right) - \frac{\mathcal{C}}{{i_0}^2 z} - i_0$$

$$\dots$$

$$l \geq m - \frac{\mathcal{C}}{z}\left(\frac{\pi^2}{6} - 1\right) - 10\sqrt[3]{\frac{m}{z}}$$

Proof

$$l \geq m - \frac{\mathcal{C}}{z}\sum_{i=1}^{\infty}\left(\frac{1}{i^2} - \frac{1}{i(i+1)}\right) - \frac{\mathcal{C}}{{i_0}^2 z} - i_0$$

$$\dots$$

$$l \geq m - \frac{\mathcal{C}}{z}\left(\frac{\pi^2}{6} - 1\right) - 10\sqrt[3]{\frac{m}{z}}$$

The total cost incurred by $\mathcal{OPT}$ is at least

$$\mathcal{C}_{\mathcal{OPT}}(\sigma) = l + \mathcal{C} \geq m + \mathcal{C} - \frac{\mathcal{C}}{z}\left(\frac{\pi^2}{6} - 1\right) - \mathcal{O}(\sqrt[3]{m})$$

Proof

- If $z > \frac{\pi^2}{6} - 1$
  Choosing $\mathcal{C} = 0$ we obtain that

$$\mathcal{C}_{\mathcal{OPT}}(\sigma) \geq m - \mathcal{O}(\sqrt[3]{m})$$

  and **LD**(z) is $(1 + z)$-competitive.

Proof

- If $z > \frac{\pi^2}{6} - 1$
  Choosing $\mathcal{C} = 0$ we obtain that

$$\mathcal{C}_{\mathcal{OPT}}(\sigma) \geq m - \mathcal{O}(\sqrt[3]{m})$$

  and **LD**(z) is $(1+z)$-competitive.

- If $z \leq \frac{\pi^2}{6} - 1$
  Choosing $\mathcal{C} = zm$ we obtain that

$$\mathcal{C}_{\mathcal{OPT}}(\sigma) \geq (2 + z - \frac{\pi^2}{6})m - \mathcal{O}(\sqrt[3]{m})$$

  and **LD**(z) achieves $(1+z)/(2+z-\frac{\pi^2}{6})$ competitive ratio.

Proof

Second case : $\mathcal{C} > zm$

The only difference in this case is that there are main packets k with $k \geq \lfloor \frac{\mathcal{C}}{2z} \rfloor$ and $k < m$ because $\mathcal{C} < m \leq 2zm$ since $z \geq 1/2$. Thus the number of acknowledgements sent by $\mathcal{OPT}$ is

$$l \geq (m - \lfloor \frac{\mathcal{C}}{2z} \rfloor)\frac{1}{2} - \frac{1}{2} + \sum_{i=3}^{i_0} \Big((\lfloor \frac{\mathcal{C}}{(i-1)z} \rfloor - \lfloor \frac{\mathcal{C}}{iz} \rfloor)\frac{1}{i} - \frac{1}{i} \Big)$$

$$\geq \ldots \geq$$

$$\geq \frac{1}{2}m - \frac{\mathcal{C}}{z}\Big(\frac{\pi^2}{6} - 1.5\Big) - 10\sqrt[3]{\frac{m}{z}}$$

Proof

Thus the optimum cost is at least

$$\mathcal{C}_{\mathcal{OPT}}(\sigma) = l + \mathcal{C} \geq \frac{1}{2}m + \mathcal{C} - \frac{\mathcal{C}}{z}\left(\frac{\pi^2}{6} - 1.5\right) - \mathcal{O}(\sqrt[3]{m})$$

Note that $z \geq 1/2 > (\frac{\pi^2}{6} - 1.5)$. Since $\mathcal{C} > zm$, we obtain

$$\mathcal{C}_{\mathcal{OPT}}(\sigma) \geq (2 + z - \frac{\pi^2}{6})m - \mathcal{O}(\sqrt[3]{m})$$

and **LD**(z) achieves $(1+z)/(2+z-\frac{\pi^2}{6})$ competitive ratio. $\square$

Layout

Theorem 2.2

Let A be a deterministic online algorithm. If A is c-competitive, then $c \geq \pi^2/6$.

Theorem 2.2

Let A be a deterministic online algorithm. If A is c-competitive, then $c \geq \pi^2/6$.

Proof

We construct a family of request sequences σ_l, for any $l \geq 8$. For a fixed l in this range, let $i_0 = \lfloor \sqrt[3]{l} \rfloor - 2$ and $l' = \lfloor \frac{l}{i_0+1} \rfloor$. We number the packets in σ_l starting with l'. Packet l' is sent at time 0.

For any i with $l' < i \leq l$, packet i is sent exactly $(\frac{\pi^2}{6} - 1)i$ time units after packet $i-1$. For any i with $i > l$, packet i is sent exactly $(\frac{\pi^2}{6} - 1)l$ time units after packet $i-1$.

The adversary stops sending packets as soon as the online algorithm acknowledges an incoming packet together with the preceding packet.

Let m be the number of the last packet sent by the adversary.

- If $m \leq l$
  The adversary can acknowledge each packet immediately and its cost is $\mathcal{C}_{\mathcal{ADV}}(\sigma_l) = m - l' + 1$.
  The online algorithm A serves the first $m - l' - 1$ packets with seperate acknowledgements and the last two packets with a joint acknowledgement. The total online cost is at least

$$\mathcal{C}_A(\sigma_l) = m - l' + (\frac{\pi^2}{6} - 1)m = \frac{\pi^2}{6}(m - \frac{6}{\pi^2}l')$$

$$\geq \frac{\pi^2}{6}(m - l' + 1) = \frac{\pi^2}{6}\mathcal{C}_{ADV}(\sigma_l)$$

  The last inequality holds because $l' \geq \frac{\pi^2}{6}/(\frac{\pi^2}{6} - 1)$, for $l \geq 8$.

Lemma 2.3

Let $1 \leq i \leq i_0$. An acknowledgement interval that ends after the arrival of packet $\lfloor \frac{l}{i} \rfloor$ must have started after the arrival of packet $\lfloor \frac{l}{i+1} \rfloor$.

Lemma 2.3

Let $1 \leq i \leq i_0$. An acknowledgement interval that ends after the arrival of packet $\lfloor \frac{l}{i} \rfloor$ must have started after the arrival of packet $\lfloor \frac{l}{i+1} \rfloor$.

- If $m > l$
  The adversary chooses acknowledgement intervals of length $(\frac{\pi^2}{6} - 1)l$.
  To estimate the total number of acknowledgements incurred by the adversary we use the following charging scheme, which is similar to the previous.

Lemma 2.3

Let $1 \leq i \leq i_0$. An acknowledgement interval that ends after the arrival of packet $\lfloor \frac{l}{i} \rfloor$ must have started after the arrival of packet $\lfloor \frac{l}{i+1} \rfloor$.

- If $m > l$
  The adversary chooses acknowledgement intervals of length $(\frac{\pi^2}{6} - 1)l$.
  To estimate the total number of acknowledgements incurred by the adversary we use the following charging scheme, which is similar to the previous.
  If an acknowledgement interval contains i packets, then the cost of 1 is distributed evenly among the packets.
  An acknowledgement interval that ends no later than the arrival of packet $\lfloor \frac{l}{i} \rfloor$, $1 \leq i \leq i_0$, contains at least $i + 1$ packets.

Hence, packets k with $\lfloor \frac{l}{i+1} \rfloor < k \leq \lfloor \frac{l}{i} \rfloor$ are charged a cost of at most $\lfloor \frac{1}{i+1} \rfloor$.

Hence, packets k with $\lfloor \frac{l}{i+1} \rfloor < k \leq \lfloor \frac{l}{i} \rfloor$ are charged a cost of at most $\lfloor \frac{1}{i+1} \rfloor$.

However a packet k, with $\lfloor \frac{l}{i+1} \rfloor < k \leq \lfloor \frac{l}{i} \rfloor$, may be contained in an acknowledgement interval that ends after the arrival of packet $\lfloor \frac{l}{i} \rfloor$. By Lemma 2.3, such an acknowledgement interval cannot end after the arrival of packet $\lfloor \frac{l}{i-1} \rfloor$, if $i \geq 2$. Thus, packet k is assigned a cost of $\frac{1}{i}$.

Hence, packets k with $\lfloor \frac{l}{i+1} \rfloor < k \leq \lfloor \frac{l}{i} \rfloor$ are charged a cost of at most $\lfloor \frac{1}{i+1} \rfloor$.
However a packet k, with $\lfloor \frac{l}{i+1} \rfloor < k \leq \lfloor \frac{l}{i} \rfloor$, may be contained in an acknowledgement interval that ends after the arrival of packet $\lfloor \frac{l}{i} \rfloor$. By Lemma 2.3, such an acknowledgement interval cannot end after the arrival of packet $\lfloor \frac{l}{i-1} \rfloor$, if $i \geq 2$. Thus, packet k is assigned a cost of $\frac{1}{i}$.
There are at most $i+1$ packets that have this cost because each packet k in the latter range has a distance of at least $(\frac{\pi^2}{6} - 1)(\lfloor \frac{l}{i+1} \rfloor + 1) > (\frac{\pi^2}{6} - 1)\frac{l}{i+1}$ to its preceding packet.

The total cost charged to all of the packets is upper bounded by

$$(m-l-1)\frac{1}{2}+1+\sum_{i=1}^{i_0}\Big((\lfloor\frac{l}{i}\rfloor-\lfloor\frac{l}{i+1}\rfloor)\frac{1}{i+1}+(i+1)(\frac{1}{i}-\frac{1}{i+1})\Big)+\frac{1}{i_0+1}$$

$$\leq \ldots \leq$$

$$\leq \frac{m}{2}+\frac{l}{2}-l\Big(\frac{\pi^2}{6}-1\Big)+\mathcal{O}(\log l)$$

The total cost charged to all of the packets is upper bounded by

$$(m-l-1)\frac{1}{2}+1+\sum_{i=1}^{i_0}\Big((\lfloor\frac{l}{i}\rfloor-\lfloor\frac{l}{i+1}\rfloor)\frac{1}{i+1}+(i+1)(\frac{1}{i}-\frac{1}{i+1})\Big)+\frac{1}{i_0+1}$$

$$\leq \ldots \leq$$

$$\leq \frac{m}{2}+\frac{l}{2}-l\Big(\frac{\pi^2}{6}-1\Big)+\mathcal{O}(\log l)$$

Since the maximum acknowledgement delay incurred by the adversary is $(\frac{\pi^2}{6}-1)l$, its total cost is $\frac{1}{2}(m+l)+\mathcal{O}(\log l)$.

The total cost incurred by the online algorithm A is $m - l' + (\frac{\pi^2}{6} - 1)l$.

The total cost incurred by the online algorithm A is $m - l' + (\frac{\pi^2}{6} - 1)l$. We conclude that the ratio of the online cost to the adversary's cost is

$$\frac{\frac{\pi^2}{6}l + m - l - l'}{l + \frac{1}{2}(m - l) + \mathcal{O}(\log l)}$$

Since $l' = o(l)$ and $\mathcal{O}(\log l) = o(l)$, this ratio approaches a value of at least $\frac{\pi^2}{6}$ as $l \to \infty$, no matter how the online algorithm chooses m, $m > l$. $\square$

For any integer $p \geq 1$, we want to minimize

$$f_p = m + \max_{1 \leq i \leq m} {d_i}^p$$

For any integer $p \geq 1$, we want to minimize

$$f_p = m + \max_{1 \leq i \leq m} {d_i}^p$$

Let

$$c_p = 1 + \sum_{q=1}^{p+1} (-1)^{p+1-q} \zeta(q)$$

Where $\zeta(p) = \sum_{i=1}^{\infty} \frac{1}{i^p}$, for any $p \geq 2$, known as the Riemann zeta function. We define $\zeta(1) := 1$.

Let $g(p) = \sum_{i=1}^{\infty} \frac{1}{i^p(i+1)}$. Then, for $p \geq 2$,

$$g(p) = \sum_{i=1}^{\infty} \frac{1}{i^p(i+1)} = \sum_{i=1}^{\infty} \frac{1}{i^p} - \sum_{i=1}^{\infty} \frac{1}{i^{p-1}(i+1)}$$

$$= \zeta(p) - g(p-1)$$

Let $g(p) = \sum_{i=1}^{\infty} \frac{1}{i^p(i+1)}$. Then, for $p \geq 2$,

$$g(p) = \sum_{i=1}^{\infty} \frac{1}{i^p(i+1)} = \sum_{i=1}^{\infty} \frac{1}{i^p} - \sum_{i=1}^{\infty} \frac{1}{i^{p-1}(i+1)}$$

$$= \zeta(p) - g(p-1)$$

Applying this reccurence repeatedly we obtain $g(p) = \sum_{q=1}^{p} (-1)^{p-q} \zeta(q)$. Thus, $c_p = 1 + g(p+1)$.

Let $g(p) = \sum_{i=1}^{\infty} \frac{1}{i^p(i+1)}$. Then, for $p \geq 2$,

$$g(p) = \sum_{i=1}^{\infty} \frac{1}{i^p(i+1)} = \sum_{i=1}^{\infty} \frac{1}{i^p} - \sum_{i=1}^{\infty} \frac{1}{i^{p-1}(i+1)}$$

$$= \zeta(p) - g(p-1)$$

Applying this reccurence repeatedly we obtain $g(p) = \sum_{q=1}^{p} (-1)^{p-q} \zeta(q)$.
Thus, $c_p = 1 + g(p+1)$.
Further, we have $g(p+1) = \frac{1}{2} + \sum_{i=2}^{\infty} \frac{1}{i^{p+1}(i+1)}$, where the last sum is always positive and tends to 0 as $p \to \infty$.
We conclude that c_p is decreasing in p and tends to 1.5 as $p \to \infty$.

p	c_p
1	1.6449
2	1.5571
3	1.5252
4	1.5117
5	1.5056
6	1.5027
7	1.5013
8	1.5007
9	1.5003
10	1.5002

Table 1: Some values of c_p

Layout

We generalize the previous algorithm. Let z be a positive real number.

Algorithm Delay(z, p)

Set the initial delay to $d = \sqrt[p]{z}$ and send out the first acknowledgement at time $a_1 + d$. In general, assume that i acknowledgements have been sent and that j packets have been processed so far. Set $d = \sqrt[p]{(i+1)z}$ and send the $(i+1)$-st acknowledgement at time $a_{j+1} + d$.

Theorem 3.1

Setting $z_p = c_p - 1$, the algorithm Delay(z_p, p) is c_p-competitive.

Proof

In the following we call the online algorithm $D(z_p, p)$ for short.

- Suppose that the online algorithm serves the input sequence using m ackowledgements. Then, its total cost is $\mathcal{C}_{D(z_p,p)}(\sigma) = m + (\sqrt[p]{mz_p})^p = (1 + z_p)m = c_p m$.

Proof

In the following we call the online algorithm $D(z_p, p)$ for short.

- Suppose that the online algorithm serves the input sequence using m ackowledgements. Then, its total cost is $\mathcal{C}_{D(z_p,p)}(\sigma) = m + (\sqrt[p]{mz_p})^p = (1 + z_p)m = c_p m$.

- Let $\mathcal{C}$ be the maximum acknowledgement delay incurred by the optimum offline algorithm $\mathcal{OPT}$. If $\mathcal{C} > \sqrt[p]{m}$, then the optimum offline cost is at least m and $D(z_p, p)$ is clearly c_p-competitive. Therefore we assume $\mathcal{C} \leq \sqrt[p]{m}$.

Proof

We analyze the optimum offline cost using the same terms and charging scheme we used in the proof of Theorem 2.1.

- We number the m main packets in the input from 0 to $m-1$.
- Let $i_0 = \lfloor \sqrt[2p+1]{\mathcal{C}^p/z_p} \rfloor - 1$. We assume $i_0 \geq 4$. If $i_0 \leq 3$, $\mathcal{C}$ is upper bounded by a constant and we can easily conclude that $D(z_p, p)$ is c_p-competitive.

Proof

First case : $\mathcal{C} < \sqrt[p]{z_p m}$

- If an acknowledgement interval starting at or after the arrival of main packet $\lfloor \frac{\mathcal{C}^p}{i^p z_p} \rfloor$ can contain at most i main packets.
- An acknowledgement interval containing main packet k, with $k \geq \lfloor \frac{\mathcal{C}^p}{i^p z_p} \rfloor$ must have started after packet $\lfloor \frac{\mathcal{C}^p}{(i+1)^p z_p} \rfloor$.

We conclude, using the same arguments as in the proof of Theorem 2.1, that the number l of acknowledgements sent by $\mathcal{OPT}$ is at least ...

Proof

$$l \geq m - \lfloor \frac{\mathcal{C}^p}{z_p} \rfloor - 1 + \sum_{i=2}^{i_0} \Big((\lfloor \frac{\mathcal{C}^p}{(i-1)^p z_p} \rfloor - \lfloor \frac{\mathcal{C}^p}{i^p z_p} \rfloor) \frac{1}{i} - \frac{1}{i} \Big)$$

$$\dots$$

$$l \geq m - \frac{\mathcal{C}^p}{z_p} z_p - \frac{\mathcal{C}^p}{{i_0}^{p+1} z_p} - H_{i_0}$$

Here H_{i_0} denotes the i_0-th Harmonic number.

Lemma 3.1

The term $\frac{\mathcal{C}^p}{{i_0}^{p+1} z_p}$ is $o(m)$

Proof

Using the above Lemma we obtain that the number of acknowledgements sent by $\mathcal{OPT}$ is at least $m - \mathcal{C}^p - o(m)$ and the total cost is $\mathcal{C}_{\mathcal{OPT}}(\sigma) \geq m - o(m)$. This implies that $D(z_p, p)$ is c_p-competitive.

Proof

Using the above Lemma we obtain that the number of acknowledgements sent by $\mathcal{OPT}$ is at least $m - \mathcal{C}^p - o(m)$ and the total cost is $\mathcal{C}_{\mathcal{OPT}}(\sigma) \geq m - o(m)$. This implies that $D(z_p, p)$ is c_p-competitive.

Second case : $\mathcal{C} \geq \sqrt[p]{z_p m}$

In this case, since $\mathcal{C}$ is large, there are not necessarily main packets k with $k \geq \lfloor \frac{\mathcal{C}^p}{z_p} \rfloor$ and $\mathcal{C} < m$.

However there are packets $k \geq \lfloor \frac{\mathcal{C}^p}{2^p z_p} \rfloor$ and $k < m$ because $\mathcal{C}^p/(2^p z_p) < m$ is equivalent to $\mathcal{C} \leq \sqrt[p]{m 2^p z_p}$ and this holds because $\mathcal{C} \leq \sqrt[p]{m}$ and $z_p = g(p+1) = \frac{1}{2} + \sum_{i=2}^{\infty} \frac{1}{i^{p+1}(i+1)} > \frac{1}{2}$.

Proof

Thus the number of ackowledgements l sent by $\mathcal{OPT}$ is at least

$$l \geq \left(m - \lfloor \frac{\mathcal{C}^p}{2^p z_p} \rfloor\right)\frac{1}{2} - \frac{1}{2} + \sum_{i=3}^{i_0}\left((\lfloor \frac{\mathcal{C}^p}{(i-1)^p z_p} \rfloor - \lfloor \frac{\mathcal{C}^p}{i^p z_p} \rfloor)\frac{1}{i} - \frac{1}{i}\right)$$

$$\cdots$$

$$l \geq \frac{m}{2} - \frac{\mathcal{C}^p}{z_p}(z_p - \frac{1}{2}) - o(m)$$

Proof

Thus the number of ackowledgements l sent by $\mathcal{OPT}$ is at least

$$l \geq \left(m - \lfloor \frac{\mathcal{C}^p}{2^p z_p} \rfloor\right)\frac{1}{2} - \frac{1}{2} + \sum_{i=3}^{i_0}\left((\lfloor \frac{\mathcal{C}^p}{(i-1)^p z_p} \rfloor - \lfloor \frac{\mathcal{C}^p}{i^p z_p} \rfloor)\frac{1}{i} - \frac{1}{i}\right)$$

$$\dots$$

$$l \geq \frac{m}{2} - \frac{\mathcal{C}^p}{z_p}(z_p - \frac{1}{2}) - o(m)$$

We conclude that the optimum offline cost is at least $\mathcal{C}_{\mathcal{OPT}}(\sigma) \geq \frac{m}{2} + \frac{\mathcal{C}^p}{2z_p} - o(m) \geq m - o(m)$ because $\mathcal{C} \geq \sqrt[p]{z_p m}$ and $D(z_p, p)$ is c_p-competitive. $\square$

Layout

Theorem 3.2

Let A be a deterministic online algorithm. If A is c-competitive, then $c \geq c_p$.

Theorem 3.2

Let A be a deterministic online algorithm. If A is c-competitive, then $c \geq c_p$.

Proof

We construct a family of request sequences σ_l, for any $l \geq 1$. For a fixed l, let $i_0 = \lfloor \sqrt[2p+1]{l} \rfloor - 1$ and $l' = \lfloor \frac{l}{i_0+1^p} \rfloor$. We number the packets in σ_l starting with l'. Packet l' is sent at time 0.

Theorem 3.2

Let A be a deterministic online algorithm. If A is c-competitive, then $c \geq c_p$.

Proof

We construct a family of request sequences σ_l, for any $l \geq 1$. For a fixed l, let $i_0 = \lfloor \sqrt[2p+1]{l} \rfloor - 1$ and $l' = \lfloor \frac{l}{i_0+1^p} \rfloor$. We number the packets in σ_l starting with l'. Packet l' is sent at time 0.

- For any k with $l' < k \leq l$, packet k is sent $\sqrt[p]{z_p k}$ time units after packet $k-1$.

Theorem 3.2

Let A be a deterministic online algorithm. If A is c-competitive, then $c \geq c_p$.

Proof

We construct a family of request sequences σ_l, for any $l \geq 1$. For a fixed l, let $i_0 = \lfloor \sqrt[2p+1]{l} \rfloor - 1$ and $l' = \lfloor \frac{l}{i_0+1^p} \rfloor$. We number the packets in σ_l starting with l'. Packet l' is sent at time 0.

- For any k with $l' < k \leq l$, packet k is sent $\sqrt[p]{z_p k}$ time units after packet $k-1$.
- For $k > l$, packets k and $k-1$ are separated by exactly $\sqrt[p]{z_p l}$ time units .

Proof

The adversary stops sending packets as soon as the online algorithm acknowledges an incoming packet together with the preceding packet. Let m be the number of the last packet sent.

Proof

The adversary stops sending packets as soon as the online algorithm acknowledges an incoming packet together with the preceding packet. Let m be the number of the last packet sent.

- If $m \leq l$
  - The cost incurred by the online algorithm A is at least $m - l' + (z_p m)^{\frac{p}{p}} = c_p m - l'$.
  - The cost incurred by the adversary is at most $m - l' + 1$.

Proof

The adversary stops sending packets as soon as the online algorithm acknowledges an incoming packet together with the preceding packet. Let m be the number of the last packet sent.

- If $m \leq l$
  - The cost incurred by the online algorithm A is at least $m - l' + (z_p m)^{\frac{p}{p}} = c_p m - l'$.
  - The cost incurred by the adversary is at most $m - l' + 1$.

  The ratio of the cost incurred by A to the cost incurred by the adversary is at least ...

Proof

$$\frac{c_p m - l'}{m - l' + 1} = c_p + \frac{z_p m' - c_p}{m - l' + 1}$$

and this expression is at least c_p if $l \geq 2^{2p+1}$.

Proof

$$\frac{c_p m - l'}{m - l' + 1} = c_p + \frac{z_p m' - c_p}{m - l' + 1}$$

and this expression is at least c_p if $l \geq 2^{2p+1}$.

- If $m > l$ The adversary chooses an acknowledgement interval of $\sqrt[p]{z_p m}$ time units.
  Using the familiar charging scheme we conclude that the total number of acknowledgements sent by the adversary is at most ...

Proof

$$(m-l)\frac{1}{2}+\frac{1}{2}+\sum_{i=1}^{i_0}\left((\lfloor\frac{l}{i^p}\rfloor-\lfloor\frac{l}{i+1^p}\rfloor)\frac{1}{i+1}+\frac{1}{i}\right)+\frac{1}{i_0+1}$$

$$\cdots$$

$$\leq (m-l)\frac{1}{2}+l-z_p l+\mathcal{O}(\log l)$$

Proof

$$(m-l)\frac{1}{2}+\frac{1}{2}+\sum_{i=1}^{i_0}\left((\lfloor\frac{l}{i^p}\rfloor-\lfloor\frac{l}{i+1^p}\rfloor)\frac{1}{i+1}+\frac{1}{i}\right)+\frac{1}{i_0+1}$$

$$\dots$$

$$\leq (m-l)\frac{1}{2}+l-z_p l+\mathcal{O}(\log l)$$

The total cost paid by the adversary is at most $(m-l)\frac{1}{2}+l+\mathcal{O}(\log l)$ and the ratio of the cost incurred by A to the cost incurred by the adversary is at least

$$\frac{c_p l+m-l'-l}{(m-l)\frac{1}{2}+l+\mathcal{O}(\log l)}$$

This ratio approaches a value of at least c_p as $l\to\infty$ because $l'=o(l)$. $\square$

Layout

Theorem 4.1

For the dynamic TCP acknowledgement problem with objective function f, no randomized online algorithm can achieve a competitive ratio smaller than $c \geq 3/(3 - \frac{2}{e})$ against any oblivious adversary.

Theorem 4.1

For the dynamic TCP acknowledgement problem with objective function f, no randomized online algorithm can achieve a competitive ratio smaller than $c \geq 3/(3 - \frac{2}{e})$ against any oblivious adversary.

Proof

We apply Yao's principle and construct a probability distribution on input sequences σ_l, for any integer $l \geq 1$, such that for any deterministic online algorithm D,

$$\lim_{l \to \infty} \frac{E[\mathcal{C}_D(\sigma_l)]}{E[\mathcal{C}_{ADV}(\sigma_l)]} \geq \frac{3}{3 - 2/e}$$

and

$$\lim_{l \to \infty} E[\mathcal{C}_{ADV}(\sigma_l)] = \infty$$

Proof

- An input σ_l consists of triples.
- A triple is a set of three data packets that are separated by l time units each.
- The adversary sends triples with a large time distance between them.
- With probability $p_i = q(1-q)^{i-1}$, where $q = 1/l$, the adversary sends exactly i triples, for any $i \geq 1$.
- Triple i and $i+1$ are separated by $3l/p_{i+1}$ time units.

Proof

- First case : the deterministic online algorithm acknowledges packets from different triples together.
  - If this happens for the first time for packets from triples i and $i+1$, then the expected cost is at least $p_{i+1}(3l/p_{i+1}) = 3l$.

Proof

- First case : the deterministic online algorithm acknowledges packets from different triples together.
  - If this happens for the first time for packets from triples i and $i+1$, then the expected cost is at least $p_{i+1}(3l/p_{i+1}) = 3l$.
- Second case : the deterministic online algorithm never acknowledges packets from different triples together.
  - We characterize an algorithm by two integers $l_1, l_2 \geq 0$, with $l_1 < l_2$.
  - $l_1 + 1$ is the first triple where the algorithm acknowledges at least two packets together.
  - $l_2 + 1$ is the first triple where all the three packets are acknowledged together.

Proof

We refer to this strategy as $D(l_1, l_2)$, $l_1 \leq l_2$.

- Algorithm $D(l_1, \infty)$, $l_1 \geq 0$, never acknowledges all the three packets of one triple together.
- Algorithm $D(\infty, \infty)$ never acknowledges any packets together.

Proof

We refer to this strategy as $D(l_1, l_2)$, $l_1 \leq l_2$.

- Algorithm $D(l_1, \infty)$, $l_1 \geq 0$, never acknowledges all the three packets of one triple together.
- Algorithm $D(\infty, \infty)$ never acknowledges any packets together.

Lemma 4.1

a) If $l_1 < l_2$, then

$$E[\mathcal{C}_{D(l_1,l_2)}(\sigma_l)] = E[\mathcal{C}_{D(l_1+1,l_2)}(\sigma_l)]$$

b) If $l_1 \leq l_2$, then

$$E[\mathcal{C}_{D(l_1,l_2)}(\sigma_l)] = E[\mathcal{C}_{D(l_1,l_2+1)}(\sigma_l)]$$

c) For any $l_1 \geq 0$, then

$$E[\mathcal{C}_{D(l_1,\infty)}(\sigma_l)] = E[\mathcal{C}_{D(l_1+1,\infty)}(\sigma_l)]$$

Proof

- Parts a) and b) of the Lemma 4.1 imply that $E[\mathcal{C}_{D(l_1,l_2)}(\sigma_l)] = E[\mathcal{C}_{D(0,0)}(\sigma_l)]$ for any $0 \leq l_1 \leq l_2$.
  Hence it suffices to compute
  $E[\mathcal{C}_{D(0,0)}(\sigma_l)] = \sum_{i=1}^{\infty}(2l+i)p_i = 2l + 1/q = 3l$.

Proof

- Parts a) and b) of the Lemma 4.1 imply that $E[\mathcal{C}_{D(l_1,l_2)}(\sigma_l)] = E[\mathcal{C}_{D(0,0)}(\sigma_l)]$ for any $0 \leq l_1 \leq l_2$.
  Hence it suffices to compute
  $E[\mathcal{C}_{D(0,0)}(\sigma_l)] = \sum_{i=1}^{\infty}(2l+i)p_i = 2l + 1/q = 3l$.
- Part c) implies $E[\mathcal{C}_{D(0,\infty)}(\sigma_l)] = E[\mathcal{C}_{D(l_1,\infty)}(\sigma_l)]$, for any $l_1 \geq 0$.
  We have $E[\mathcal{C}_{D(0,\infty)}(\sigma_l)] = \sum_{i=1}^{\infty}(l+2i)p_i = 3l$.

Proof

- Parts a) and b) of the Lemma 4.1 imply that $E[\mathcal{C}_{D(l_1,l_2)}(\sigma_l)] = E[\mathcal{C}_{D(0,0)}(\sigma_l)]$ for any $0 \leq l_1 \leq l_2$.
  Hence it suffices to compute
  $E[\mathcal{C}_{D(0,0)}(\sigma_l)] = \sum_{i=1}^{\infty}(2l+i)p_i = 2l + 1/q = 3l$.
- Part c) implies $E[\mathcal{C}_{D(0,\infty)}(\sigma_l)] = E[\mathcal{C}_{D(l_1,\infty)}(\sigma_l)]$, for any $l_1 \geq 0$.
  We have $E[\mathcal{C}_{D(0,\infty)}(\sigma_l)] = \sum_{i=1}^{\infty}(l+2i)p_i = 3l$.
- Finally, $E[\mathcal{C}_{D(\infty,\infty)}(\sigma_l)] = \sum_{i=1}^{\infty} 3ip_i = 3l$.

Thus, in any case the expected online cost is at least $3l$.

Proof

It remains to analyze the expected cost incurred by the adversary.
If the input consists of at most l triples, the adversary acknowledges the packets individually.
Otherwise it incurs a delay of $2l$ and acknowledges the packets of each triple together. So,

Proof

It remains to analyze the expected cost incurred by the adversary.
If the input consists of at most l triples, the adversary acknowledges the packets individually.
Otherwise it incurs a delay of $2l$ and acknowledges the packets of each triple together. So,

$$E[C_{ADV}(\sigma_l)] = \sum_{i=1}^{l} 3ip_i + \sum_{i=l+1}^{\infty} p_i(i+2l)$$

$$\dots$$

$$= 3l - 2l(1-1/l)^l$$

Proof

It remains to analyze the expected cost incurred by the adversary.
If the input consists of at most l triples, the adversary acknowledges the packets individually.
Otherwise it incurs a delay of $2l$ and acknowledges the packets of each triple together. So,

$$E[\mathcal{C}_{ADV}(\sigma_l)] = \sum_{i=1}^{l} 3ip_i + \sum_{i=l+1}^{\infty} p_i(i+2l)$$

$$\cdots$$

$$= 3l - 2l(1-1/l)^l$$

Thus, $\lim_{l\to\infty} E[\mathcal{C}_{ADV}(\sigma_l)]/l = 3 - 2/e$ and the Theorem follows. $\square$

Theorem 4.2

For the dynamic TCP acknowledgement problem with objective function f_p, no randomized online algorithm can achieve a competitive ratio smaller than $c \geq 2/(2 - \frac{1}{e})$ against any oblivious adversary.

Theorem 4.2

For the dynamic TCP acknowledgement problem with objective function f_p, no randomized online algorithm can achieve a competitive ratio smaller than $c \geq 2/(2 - \frac{1}{e})$ against any oblivious adversary.

Proof

- An input σ_l, for any integer $l \geq 1$, consists of pairs.
- A pair are two packets that are $\sqrt[p]{l}$ time units apart.
- With probability $p_i = q(1-q)^{i-1}$, where $q = 1/l$, the input consists of i pairs, for any $i \geq 1$.
- Pairs i and $i+1$ are separated by $\sqrt[p]{2l/p_{i+1}}$ time units.

Proof

- First case : the deterministic online algorithm acknowledges packets from different pairs together.
  - If this happens for the first time for packets from pairs i and $i+1$, then the expected cost is at least $p_{i+1}(\sqrt[p]{2l/p_{i+1}})^p = 2l$.

Proof

- First case : the deterministic online algorithm acknowledges packets from different pairs together.
  - If this happens for the first time for packets from pairs i and $i+1$, then the expected cost is at least $p_{i+1}(\sqrt[p]{2l/p_{i+1}})^p = 2l$.
- Second case : the deterministic online algorithm never acknowledges packets from different pairs together.
  - Denote by $D(l')$, $l' \geq 1$, the algorithm that acknowledges packets in the first l' pairs separately and the packets in the $(l'+1)$-st pair together.
  - $D(\infty)$ is the algorithm that never acknowledges packets together.

Proof

- We have $E[\mathcal{C}_{D(\infty)}(\sigma_l)] = \sum_{i=1}^{\infty} 2ip_i = 2q/q^2 = 2l.$

Proof

- We have $E[\mathcal{C}_{D(\infty)}(\sigma_l)] = \sum_{i=1}^{\infty} 2ip_i = 2q/q^2 = 2l$.
- It is proved that for any $l' \geq 0$, $E[\mathcal{C}_{D(l')}(\sigma_l)] = E[\mathcal{C}_{D(l'+1)}(\sigma_l)]$ and hence $E[\mathcal{C}_{D(0)}(\sigma_l)] = E[\mathcal{C}_{D(l')}(\sigma_l)]$.
  $E[\mathcal{C}_{D(0)}(\sigma_l)] = \sum_{i=1}^{\infty} (l+i)p_i = 2l$.

So we conclude that the expected online cost is at least $2l$.

Proof

The adversary acknowledges the packets of pairs separately if at most l pairs are sent.
Otherwise, it always acknowledges the packets of pairs together. Hence,

Proof

The adversary acknowledges the packets of pairs separately if at most l pairs are sent.
Otherwise, it always acknowledges the packets of pairs together. Hence,

$$E[\mathcal{C}_{ADV}(\sigma_l)] = \sum_{i=1}^{l} 2ip_i + \sum_{i=l+1}^{\infty} p_i(i+l)$$

$$\cdots$$

$$= 2l - l(1-1/l)^l$$

Proof

The adversary acknowledges the packets of pairs separately if at most l pairs are sent.
Otherwise, it always acknowledges the packets of pairs together. Hence,

$$E[\mathcal{C}_{ADV}(\sigma_l)] = \sum_{i=1}^{l} 2ip_i + \sum_{i=l+1}^{\infty} p_i(i+l)$$

$$\cdots$$

$$= 2l - l(1-1/l)^l$$

Thus, $\lim_{l\to\infty} E[\mathcal{C}_{ADV}(\sigma_l)]/l = 2 - 1/e$ and the Theorem follows. $\square$