Packet Bundling

Jens S. Frederiksen
University of Southern Denmark, Odense
svalle@imada.sdu.dk

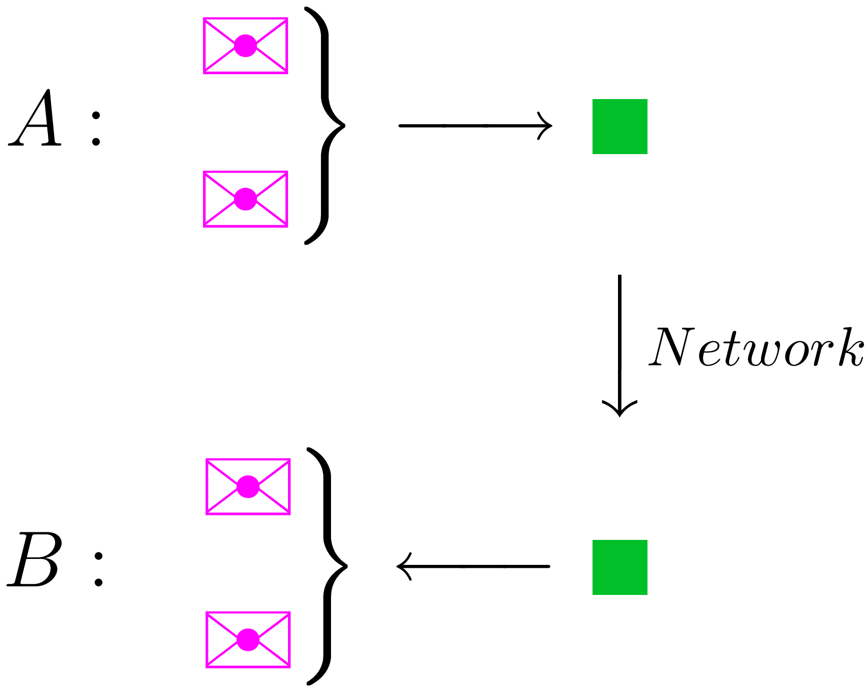
Eighth Scandinavian Workshop
on
Algorithm Theory

July 3 – 5, 2002

Joint work with Kim S. Larsen

Problem Definition

Messages ( ) are sent in packets ( ).



Messages can be bundled to reduce cost.

- $\sigma = a_1, a_2, \ldots, a_n$ messages.
- $p_1, p_2, \ldots, p_m$, packet times.
- $\tilde{p}_i$ is the set of messages contained in packet i.

Competitive Ratio

The standard quality comparison model for on-line (minimization) problems:

$$ALG(\sigma) \leq c\,OPT(\sigma) + b$$

ALG is c-competitive, if this holds for all inputs σ, and fixed constants b and c.

The competitive ratio of ALG is the infimum over all constants c for which ALG is c-competitive.

For randomized algorithms, the expected cost is used:

$$E\left[ALG(\sigma)\right] \leq c\,OPT(\sigma) + b$$

(oblivious adversary).

Previous Work

[Dooly, Goldman, Scott: STOC 1998]

Dynamic TCP Acknowledgment Problem.

Cost function

$$\eta \cdot \text{\# of packets} + (1-\eta) \sum_{\text{all messages } a} (t_{a_{\text{delivered}}} - t_{a_{\text{arrival}}})$$

or formally

$$\eta\, m + (1-\eta) \sum_{i=1}^{m} \sum_{a_j \in \tilde{p}_i} (p_i - a_j)$$

Optimal 2-competitive deterministic on-line algorithm.

[Seiden: STOC 2000]

Lower bound of $\frac{e}{e-1}$ for randomized on-line algorithms.

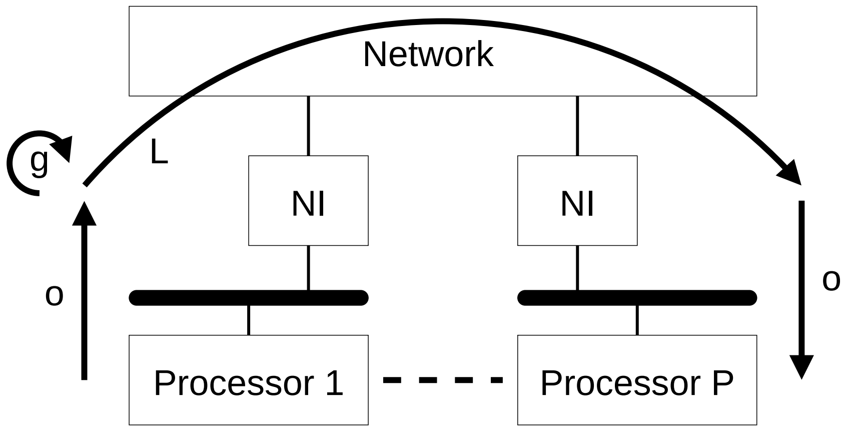
[Karlin, Kenyon, Randall: STOC 2001]

Optimal $\frac{e}{e-1}$ randomized algorithm.

LogP Model

[Culler, Karp, Patterson, Sahay, Schauser, Santos, Subramonian, von Eicken: PPOPP 1993]

LogP: a communication model for distributed computing.



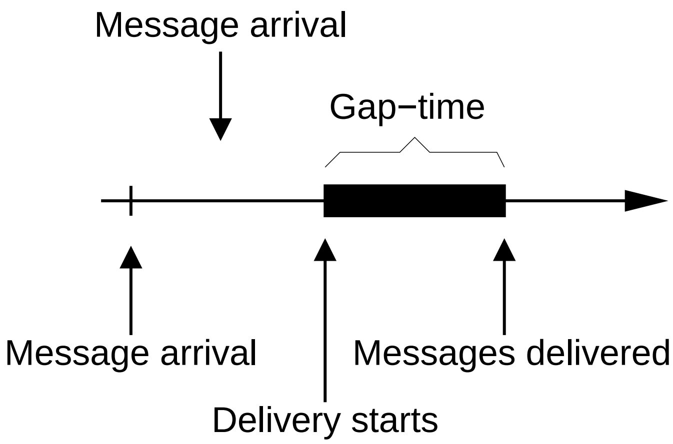
- Network latency, L.
- Processor overhead, o.
- Network gap, g.
- Number of processors, P.

Our focus: $\max\{o, g\}$.

LogP Model

The consequence of this:

We cannot start sending packets during “gap-time”



Flow-Time Cost in the LogP Model

Cost is

$$m + \sum_{i=1}^{m} \sum_{a_j \in \tilde{p}_i} \left(p_i + \max\{o, g\} - a_j \right),$$

but now we require:

$$|p_{i+1} - p_i| \geq \max\{o, g\}.$$

Results

- Any deterministic algorithm has a competitive ratio of at least 2.
- Any "uniform" randomized algorithm has a competitive ratio of at least 2.
- Any "reasonable" algorithm has a competitive ratio of at most 2.

Cost: another attempt

Measure the total time elapsed while there are

available, undelivered messages.

Formally,

$$\sum_{i=1}^{m} \left((p_i + 1) - \max\{(p_{i-1} + 1), \min_{a_j \in \tilde{p}_i} a_j\} \right)$$

Important requirement: $|p_{i+1} - p_i| \geq 1$ for all i.

Minor remarks:

- $p_0 = -\infty$
- $\max\{o, g\}$ normalized to 1.

Deterministic Algorithms

Family A_k:

A message is sent at the earliest possible time after k time units (all available messages at sending time are bundled).

Theorem The competitive ratio of A_k is:

$$\mathcal{R}(A_k) = \begin{cases} 1 + \frac{1}{1+k}, & \text{if } 0 \leq k < \hat{\varphi} \\ 1 + k, & \text{if } \hat{\varphi} \leq k \end{cases}$$

Best ratio φ is achieved by $A_{\hat{\varphi}}$.

($\varphi = \frac{1+\sqrt{5}}{2} \approx 1.618$ and $\hat{\varphi} = \varphi - 1$)

Deterministic Algorithms

Proof of Theorem

First analyze structure of worst-case sequences:

- For a worst-case sequence for A_k, we can assume that each packet contains only one message.
- There exists a worst-case sequence for A_k where, if any messages arrive during gap-time, they arrive at the beginning or the end of gap-time.

Deterministic Algorithms

Proof of Theorem

For $k \leq 1$, a worst-case sequence can be assumed to be of the following form:

- $\sigma_1 = 0$
- $\sigma_n = 0, k, k+1, k+2, \ldots, k+(n-2)$

The competitive ratio becomes:

$$\frac{A_k(\sigma_n)}{OPT(\sigma_n)} = \begin{cases} k+1, & \text{if } n = 1 \\ \frac{k+n}{k+n-1}, & \text{if } n > 1 \end{cases}$$

Maximized for either $n = 1$ or $n = 2$ (depending on k):

$$\frac{A_k(\sigma_2)}{OPT(\sigma_2)} = \frac{k+2}{k+1} = 1 + \frac{1}{1+k}$$

The proof for $k > 1$ is similar.

$A_{\hat{\varphi}}$ Is Optimal

Theorem Let ALG be any deterministic algorithm for the *Packet Bundling Problem*. Then

$$\mathcal{R}\left(ALG\right) \geq \mathcal{R}\left(A_{\hat{\varphi}}\right) = \varphi$$

Proof Construction of input sequence:

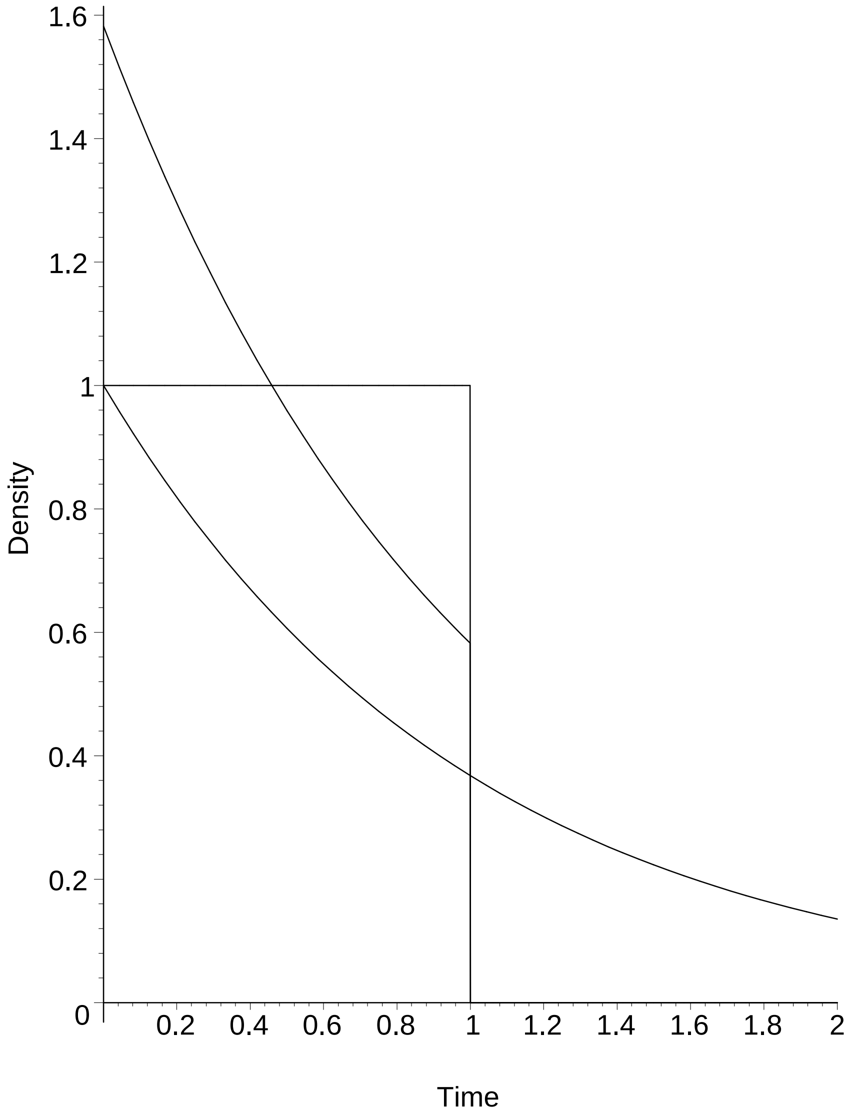
- Give one message at time 0.
- Let k be the time ALG decides to pack and send.
- If $k \leq \hat{\varphi}$, then give another message at time k (otherwise, send no more messages).

By insight from previous Theorem,

$$ALG(\sigma) = A_k(\sigma) \geq A_{\hat{\varphi}}(\sigma) = \varphi.$$

Randomized Algorithms

We can randomize this in different ways.



Uniform density gives the best competitive ratio.

Randomized Algorithms – $RAND_\Delta$

Family $RAND_\Delta$:

When a new message arrives and no other messages are waiting, this message is sent at the earliest possible time after a period of time chosen uniformly between 0 and Δ (all available messages at sending time are bundled).

Theorem The expected competitive ratio of $RAND_\Delta$ is

$$\overline{\mathcal{R}}\,(RAND_\Delta) = \begin{cases} \frac{1}{2} + \frac{3}{2(\Delta+1)} & \text{if } 0 \leq \Delta \leq \frac{1}{2} \\ \frac{6\Delta^2+4\Delta+1}{4\Delta(1+\Delta)} & \text{if } \frac{1}{2} \leq \Delta \leq \sqrt[3]{\frac{1}{2}} \\ \frac{\Delta}{2} + 1 & \text{if } \sqrt[3]{\frac{1}{2}} \leq \Delta \leq 1 \end{cases}$$

Best ratio $\frac{\sqrt[3]{\frac{1}{2}}}{2} + 1 \approx 1.397$ is achieved by $RAND_{\sqrt[3]{\frac{1}{2}}}$.

Randomized Algorithms

Proof of Theorem

Again, analyze the structure of worst-case sequences.

At the core are sequences $\sigma = 0, a_2$, where $a_2 < 1$.

Sending a_1 at δ, cost is

$k(a_2, \delta) = \delta + 1 +$

$$\begin{cases} 0 & , \text{ if } a_2 \leq \delta \\ 1 & , \text{ if } \delta < a_2 \leq \delta + 1 - \Delta \\ 1 + \frac{(a_2+\Delta)-(\delta+1)}{2} & , \text{ if } \delta + 1 - \Delta < a_2 \leq 1 \end{cases}$$

which leads to the expected ratio

$$\frac{1}{a_2 + 1} \frac{1}{\Delta} \int_0^{\Delta} k(a_2, \delta) d\delta$$

implying $a_2 = 0$ or $a_2 = \Delta$.

Concluding Remarks

For packet bundling in the LogP model:

- Flow-time cost cannot distinguish between reasonable algorithms.
- New measure based on network interface activity.
- Found an optimal deterministic algorithm.
- Found a better randomized algorithm from a natural family.

Open problem

- Is $RAND_{\sqrt[3]{\frac{1}{2}}}$ the optimal randomized algorithm?