Customized Newspaper Broadcast

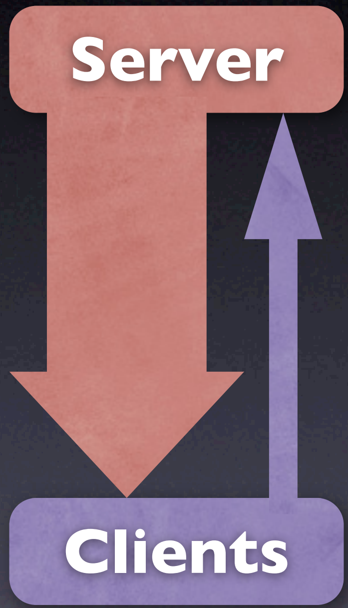
Sandeep Dey & Nicolas Schabanel

LIP - CNRS - École normale supérieure de Lyon

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

INRIA

Data Dissemination



Asymmetric media

Naturally broadcasting : wireless networks, Satellite, Multicast trees, Ethernet, Cable TV, VideoText, Mobile phones,...

Very large number of clients

Wireless internet, Hot Information Systems (traffic, football,...),...

Inefficient to serve each request individually

(overload server and network)

Data Dissemination Protocols

The server knows a ***profile*** of the clients :
the request probability for each items set

No request is sent to the server
("push" as opposed to "pull")

Data Dissemination Protocols

The server knows a ***profile*** of the clients :
the request probability for each items set

No request is sent to the server
("push" as opposed to "pull")

Example of dependencies

Each web page is ***composed of items that are shared*** by several web pages

Each page is downloaded when ***all*** its components have been received

Example of dependencies

Each web page is

composed of items

that are shared

by several web pages

Each page is
downloaded when
all its components have
been received

The problem

Instance :

n news items (components) $M_1, \dots, M_n$ of length 1

k newspaper (pages) $S_1, \dots, S_k \subseteq \{ 1, \dots, n \}$

the popularities $p_1, \dots, p_k$ of the newspapers,

$$p_1 + \cdots + p_k = 1$$

Goal : A schedule of the ***news items*** that minimizes the ***newspapers downloading time for a random user***

The problem

Instance :

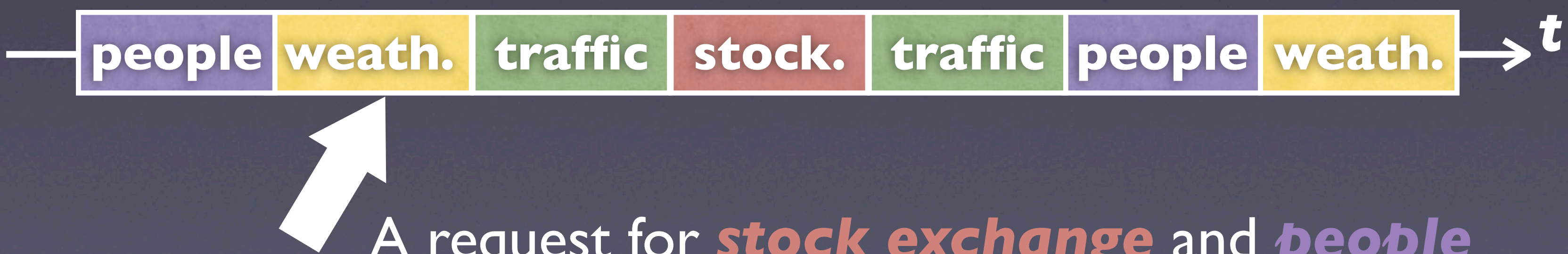
n news items (components) $M_1, \ldots, M_n$ of length **1**

k newspaper (pages) $S_1, \ldots, S_k \subseteq \{ 1, \ldots, n \}$

the popularities $p_1, \ldots, p_k$ of the newspapers,

$$p_1 + \cdots + p_k = 1$$

Goal : A schedule of the ***news items*** that minimizes the ***newspapers downloading time for a random user***



A request for ***stock exchange*** and ***people***

The problem

Instance :

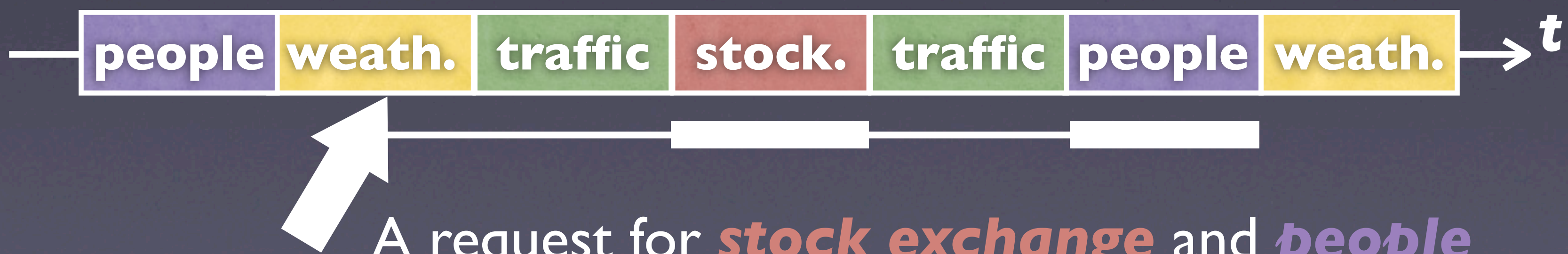
n news items (components) $M_1, \dots, M_n$ of length **1**

k newspaper (pages) $S_1, \dots, S_k \subseteq \{ 1, \dots, n \}$

the popularities $p_1, \dots, p_k$ of the newspapers,

$$p_1 + \cdots + p_k = 1$$

Goal : A schedule of the ***news items*** that minimizes the ***newspapers downloading time for a random user***



A request for ***stock exchange*** and ***people***

Previous work (1)

NP-hard without dependencies

if broadcasting a news item has a cost *(BBNS98)*
or if the news items have different lengths *(KS99, S00)*

Algorithms without dependencies

- 2-approximation, *Ammar & Wong 1985*
- 9/8-approximation, *Bar-Noy, Bhatia, Naor & Scheiber 1998*
- $(1+\varepsilon)$-approximation, *Kenyon, S. & Young 2000*

Related work : broadcast disks

by *Acharya, Alonso, Franklin & Zdonik 1995*

Previous work (2)

NP-hard with dependencies

the preemptive setting proved to be NP-hard in *[S00]* is the restriction where all newspapers are disjoint

Approximation algorithms with dependencies

- Optimal for ***two news items***, *Bar-Noy, Shilo, 2000*
- 3/2-approximation for the best ***permutation*** when ***each dependency involves at most two news items***, *Bar-Noy, Naor & Scheiber 2003*
- Experimental evidences, *Huang & Chen 2003-2004*

⇒ Taking into account dependencies ***significantly improves performances***

Cost of a schedule

$$\text{Cost(Schedule)} = \sum_{j=1}^{k} p_j \cdot \text{Avg wait(Newspaper } S_j)$$

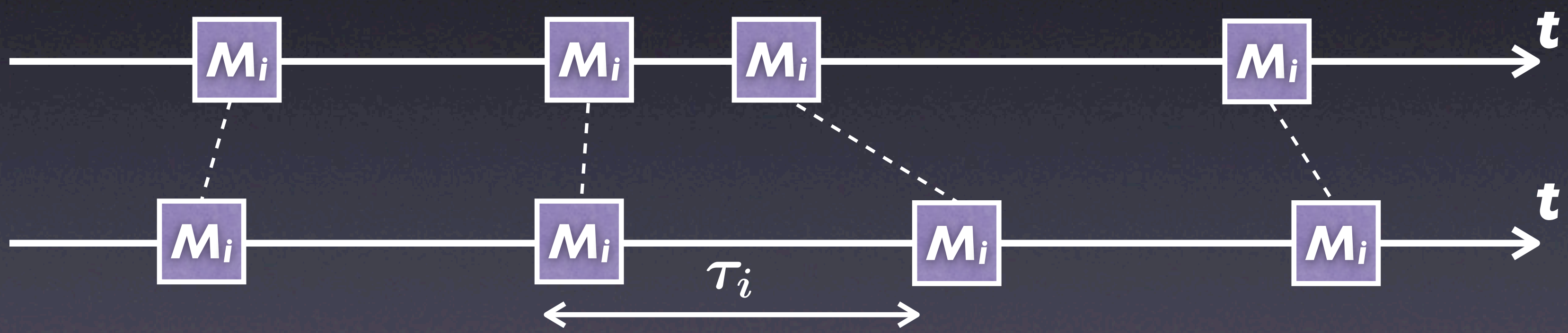
$$\text{Avg wait(Newspaper } S_j) =$$

$$\frac{1}{T}\int_{t=0}^{T} \max_{\text{News item } M_i \in S_j} \text{Wait}(M_i, t)\, dt$$

Lower bounding the cost

Convex: $\text{Avg wait(Newspaper } S_j) \geqslant \max_{\text{item } M_i \in S_j} \text{Avg wait}(M_i)$

Relaxation (AW85): *Allow overlaps and optimize the broadcasts of each news item **M_i** independently*



*Minimum when **M_i** is broadcast regularly*

$$\text{Avg wait(Newspaper } S_j) \geqslant \max_{\text{News item } M_i \in S_j} \frac{\tau_i}{2}$$

The lower bound

$$\mathrm{LB} = \begin{cases} \displaystyle \underset{\tau > 0}{\text{Minimize}} \quad \sum_{\text{Newspaper } S_j} p_j \cdot \max_{\text{News item } M_i \in S_j} \frac{\tau_i}{2} \\ \displaystyle \text{with} \quad \frac{1}{\tau_1} + \cdots + \frac{1}{\tau_n} \leqslant 1 \end{cases}$$

Convex minimization problem

Ellipsoid algorithm obtain in polynomial time a solution:

$$\tau^* \text{ such that } \mathrm{LB}(\tau^*) \leqslant \mathrm{LB} + \frac{1}{4}$$

Randomized approximation

For each time slot t :

Choose *i* in {***1***,..., ***n***} with probability $1/\tau_i^*$

Broadcast news item $\boldsymbol{M_i}$

Analysis (Coupon collector)

Each news item in newspaper ***S*** of size |***S***| is broadcast in each time slot with probability at least $1/\max_{i \in S} \tau_i^*$

The expected downloading time for ***S*** is then at most

$$\ln |S| \cdot \max_{i \in S} \tau_i^*$$

Randomized approximation

For each time slot t :

Choose *i* in {***1***,..., ***n***} with probability $1/\tau_i^*$

Broadcast news item M_i

Analysis (Coupon collector)

The expected cost for the randomized schedule is then:

$$\text{Cost}(\text{Randomized}) \leqslant \sum_{\text{Newspaper } S_j} p_j \cdot \ln |S_j| \cdot \max_{i \in J_j} \tau_i^* \leqslant 2 \ln n \cdot \text{LB}$$

⇒ The randomized scheduler is a **2 ln *n*-approximation**

The analysis is tight

A family of critical instances:

k disjoint newspapers with **ln *k*** news items each (***n* = *k* ln *k***)

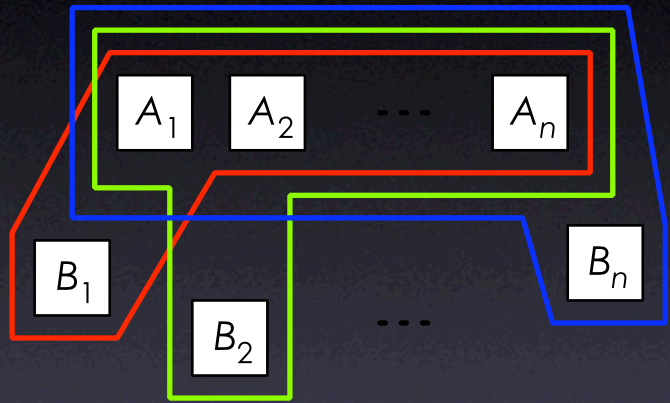
Cost(Randomized) ~ **2 ln *k*** • ***k* ln *k***

and ***OPT*** ≤ ***Cost(Round Robin)*** ~ ***k* ln *k***

thus ***Cost(Randomized)*** ~ **2 ln *n*** • ***OPT***

Previously known algorithms fail in presence of dependencies

Instance: 2*n* *items,* ***n*** newspapers equally popular as shown



Individual popularity for each item

$\boldsymbol{A_i}$ is $\sim \frac{1}{n}$ and $\boldsymbol{B_i}$ is $\sim \frac{1}{n^2}$

<u>All</u> *previous algorithms aim to broadcast* $\boldsymbol{A_i}$ *every* $\tau_A \sim n$ *and and* $\boldsymbol{A_i}$ *every* $\tau_B \sim n\sqrt{n}$ *(square-root rule, AW85)*

But for these values: $\mathrm{LB}(\tau) \sim n \cdot \frac{1}{n} \cdot \frac{\max(n, n\sqrt{n})}{2} \sim \frac{n\sqrt{n}}{2}$

while: $\mathrm{OPT} \leqslant \mathrm{Cost(Round\ Robin)} \sim \frac{n}{2}$

<u>Ignoring dependencies can induce an overhead cost factor of</u> $\sqrt{n}$ *<u>!</u>*

Perfectly periodic schedules

The cause of the bad performances of random scheduler is that news items are broadcast in ***random order***.

<u>Solution :</u> perfectly periodic schedules

Lemma *(AGH95, BNP01)*
If the τ_is are powers of 2 and verify $\frac{1}{\tau_1} + \cdots + \frac{1}{\tau_n} \leqslant 1$,
*there exists a schedule that broadcasts each **M_i** exactly every τ_i.*

Deterministic approximation

For all i, let $\beta_i = 2^\ell$, with $2^{\ell-1} < \tau_i^* \leqslant 2^\ell$

Broadcast the news items according to a perfectly periodic schedule with periods (β_i)

Analysis

Each news item ***M_i*** is broadcast every $\beta_i \leq 2\tau_i^*$

Each newspaper ***S_j*** is download after at most $2 \max_{M_i \in J_j} \tau_i^*$ time

thus,

Cost(Algorithm 2) ≤ 4 LB ≤ 4 OPT

⇒ it's a 4-approximation!

Conclusion & perspectives

Important to take into account dependencies

If not, it can induce an overhead cost factor of $\sqrt{n}$

A deterministic 4-approximation

More important:
A constant factor lower bound on *OPT*

a good estimation of ***OPT***: allows to compare *objectively* different heuristics with respect to the optimal *(and in particular to measure the real performances of a given algorithm)*

Conclusion & perspectives

Improve the approximation ratio

Faster lower bound computation

On-line or on-demand dissemination with dependencies

Some competitive algorithms exist with resource augmentation without dependencies *(EP02, BCKN05)*

Any question ?

please, email

dey.sandeep@gmail.com

nicolas.schabanel@ens-lyon.fr