سیزدهمین کنفرانس ملی انجمن کامپیوتر ایران
جزیره کیش، خلیج فارس، ایران
۱۹ الی ۲۱ اسفند ۱۳۸۶
پوستر

همه‌پخشی متحمل خطا در شبکه توری درخت‌ها تحت مدل تک‌پورت و مسیریابی غیرحساس به مسافت

فاطمه صارمی
دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف
f_saremi@ce.sharif.edu

هدی مشایخی
دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف
mashayekhi@ce.sharif.edu

محمد قدسی×
دانشکده مهندسی‌کامپیوتر، دانشگاه صنعتی شریف
پژوهشکده علوم کامپیوتر، مرکز تحقیقات فیزیک نظری و ریاضیات
ghodsi@sharif.edu

چکیده: توری درخت‌ها، شبکه‌ای با درجه محدود، قطر کم و پهنای دوبخشی زیاد می‌باشد. این شبکه‌ها می‌توانند الگوریتم‌های با الگوی ارتباطات سراسری را در زمان بهینه تحت مسیریابی غیر حساس به مسافت اجرا نمایند. رفتار توری درخت‌ها در حضور گره‌های خطادار، تاکنون تنها به صورت محدود مورد بررسی قرار گرفته است. در این مقاله ما به ارائه یک الگوریتم همه‌پخشی متحمل خطا برای این شبکه‌ها تحت مسیریابی غیر حساس به مسافت پرداخته و میزان تحمل‌پذیری خطا را درآن بررسی می‌کنیم. به طور خاص نشان می‌دهیم که توری درخت‌ها می‌تواند الگوریتم همه‌پخشی را با کاهش کارآیی خطی، با ضریب کوچک، نسبت به تعداد خطاهای موجود در شبکه و سطح گره‌های خطادار اجرا نماید.

واژه های کلیدی: توری درخت‌ها، همه‌پخشی، تحمل خطا.

۱- مقدمه

ساختار توری درخت‌ها، قطر کم خود را از درخت و پهنای دوبخشی زیاد خود را از توری اخذ کرده است. نحوه عملکرد ساختار توری درخت‌ها تحت الگوهای ارتباط سراسری بر مبنای مسیریابی غیرحساس به مسافت، در حضور خطا در گره‌های شبکه، تاکنون تنها به طور محدود مورد بررسی قرار گرفته است. اگر حد پایین پیچیدگی همه‌پخشی در گراف G تحت مدل ارتباطی M را با نماد b(M,G) نمایش دهیم، داریم: $b(H_1^N, CBT_n)=n+2=\lceil \log_2 N \rceil+1$ [۱]. T یک درخت دودویی کامل با 2^n برگ و مجموعه حداکثر S(n,b) برگ خطا دار، که $n \geq b \geq 0$، می‌باشد. S(n,b) مطابق رابطه زیر تعریف می‌شود: S(n,b) = S(n-1,b) + S(n-1,b-1) + 1، که برای هر b > 0 ، $S(n,b)=\theta(n^b)$. دراین‌صورت، می‌توان درخت دودویی کامل بدون خطای T' شامل 2^n برگ را در T تعبیه کرد بگونه‌ای که: گره‌های سطح i در T' به گره‌های بدون خطای سطح i در T ($0 \leq i \leq n$) نگاشته می‌شوند. فاکتورهای بار و تراکم در فرآیند تعبیه حداکثر 2^b بوده و، تأخیر تعبیه ۱ است [۲]. یک توری درخت‌ها با ابعاد $N \times N$ می‌تواند با روند حفظ سطح، در یک توری درخت‌ها با همان ابعاد شامل $S(\log N, b)=\theta(\log^b N)$ خطای بدترین حالت، با بار و تراکم برابر 2^{2^b}، و تأخیر برابر ۱، تعبیه شود [۲].

در [۱] نشان داده شده است ساختار توری درخت‌ها می‌تواند الگوریتم همه‌پخشی را تحت مدل ترکیب‌کننده تک‌پورتی و سوئیپینگ غیرحساس به مسافت به طور بهینه یا تقریباً بهینه اجرا کند. در این مقاله به بررسی میزان تحمل‌پذیری خطا در این الگوریتم می‌پردازیم. نشان می‌دهیم که در الگوریتم همه‌پخشی وجود خطا در گره‌های شبکه باعث افزایش تعداد دورهای الگوریتم به طور خطی نسبت، با ضریب کوچک، به تعداد خطاها و سطح گره‌های خطادار می‌گردد.

۲- تعاریف

در درخت دودویی کامل CBT_n، ($\beta=\{0,1\}$): $V(CBT_n)=\bigcup_{i=0}^{n}\beta^i$، و $\mathrm{E}(CBT_n)=\{\langle x, xa\rangle; len(x)<n \wedge a \in \beta\}$، ریشه درخت در سطح 0 و برگ ها در سطح n قرار دارند. توری مستطیلی از درخت‌ها با ارتفاعات m و n، ($MT_{m,n}$)، به صورت زیر تعریف می‌شود:

$$V(MT_{m,n})=\left(\beta^m \times \bigcup_{i=0}^{n}\beta^i\right) \bigcup \left(\bigcup_{i=0}^{m}\beta^i \times \beta^n\right)$$

$$\mathrm{E}(MT_{m,n})=\{\langle (x,y),(xa,y)\rangle; len(x)<m \wedge a \in \beta\} \bigcup \{\langle (x,y),(x,ya)\rangle; len(y)<n \wedge a \in \beta\}$$

منظور از خطا در این مقاله، عدم توانایی یک گره در ساختار توری درخت‌ها برای دریافت، ارسال یا مسیریابی یک بسته می‌باشد. در [۵] نشان داده شده است که اگر یک تعبیه G در H با فاکتور بار l، تراکم c و تأخیر d وجود داشته باشد، آنگاه H می‌تواند هر محاسباتی در G را با کاهش سرعت $O(l+c+d)$ انجام دهد. در این مقاله فرض بر این است که گره‌ها یک پورت داخلی دارند و مدل ارتباطی لینک-یک‌طرفه می‌باشد. در مدل مسیر گره‌مجزا یک گره می‌تواند یک بسته را ارسال/دریافت کند، یا به عنوان گره میانی در مسیر بسته را پیش براند. این مدل که ضعیف‌ترین مدل ارتباطی است را با نمادگذاری H_1^N نمایش می‌دهیم. پیچیدگی الگوریتم را برابر تعداد دورهای آن تعریف می‌کنیم. دور جدید وقتی آغاز می‌شود که تمامی مسیرهای ارتباطی

× این نویسنده توسط مرکز تحقیقات فیزیک نظری (قرارداد شماره...) حمایت شده است.

دورقبلی آزاد شده باشند. اگر هیچ برخوردی بین مسیرها در دورهای مختلف رخ ندهد، آنگاه به الگوریتم مستقل از دور گوییم و هر گره می‌تواند به محض پایان دور قبل، دور جدید را مستقلا آغاز نماید.

۳- همه‌پخشی متحمل خطا در توری درخت‌ها

الگوریتم پیشنهادی، بر مبنای الگوریتم همه‌پخشی ارائه شده در [۱] می‌باشد که قابلیت تحمل خطا ندارد. الگوریتم پیشنهادی که ۴ فاز دارد، در شکل ۱ آمده است.

فاز اول: با توجه به تقارن توری مربعی درخت‌ها، می‌توان گره $(0^k,0^n)$، $0 \le k \le n$ را بعنوان مبدأ همه‌پخشی در نظر گرفت. گوییم زیرمجموعه U از گره‌ها پوشانده شده اگگر هر یک از گره‌های U قبلاً پیام را دریافت کرده، و یا مسیری به گره‌ای مطلع داشته باشد Covered_by(x,s)، یعنی x توسط s پوشانده شده است. فاز ۱، یک همه‌پخشی جزئی را در درخت سطری اول انجام می‌دهد (همه‌پخشی فرد) و در پایان این فاز تمام برگ‌های درخت سطری اول پوشانده می‌شوند. اگر گره x در درخت، خطا دار و مقصد اطلاع باشد، در دوری که باید به آن گره اطلاع‌رسانی شود به گره $x0^k$، به ازای کوچکترین k که سالم باشد، خبر داده می‌شود که کار گره خطادار را در مراحل بعدی فاز ۱ انجام می‌دهد. در دور k-ام این فاز در یک درخت به ارتفاع n، نیمی از گره‌های سطح k مطلع می‌شوند. پس حداکثر 2^{n-1} گره در هر مرحله مقصد مسیرهای اطلاع‌رسانی هستند و به همین تعداد مسیر گره‌مجزا نیاز خواهیم داشت، که به صورت حریصانه از طریق سایر گره‌های شبکه فراهم می‌شود. اگر گره خطادار مسیر اطلاع‌رسانی را بسته باشد، پس از اجرای فاز ۱ در سایر گره‌ها، یک سری زیردرخت‌هایی که از درخت اصلی مجزا شده‌اند وجود دارند که هنوز اطلاع‌رسانی نشده‌اند. پس از اجرای فاز ۱ در سایر گره‌ها، با استفاده از دیگر درخت‌های شبکه، ریشه زیردرخت‌های مذکور توسط مجموعه $\mathrm{cov}ered_by$ برگ‌های درخت سطری اول مطلع می‌شوند و سپس فاز ۱ الگوریتم در این زیردرخت‌ها انجام می‌شود. در این حالت اگر گره خطادار در سطح m باشد، با فرض اینکه حداقل به اندازه تعداد این زیردرخت‌ها، گره فاقد خطا در مجموعه $\mathrm{cov}ered_by$ برگ‌های درخت سطری اول موجود باشد، تعداد دورهای اضافه شده $n-m+1$ خواهد بود. در غیراینصورت اگر تعداد زیردرخت‌های مجزا شده k، و تعداد گره‌های فاقد خطا در مجموعه $\mathrm{cov}ered_by$ برگ‌های درخت سطری اول، j باشد، تعداد دورها $\lceil k/j \rceil (n-m+1)$ افزایش خواهد یافت. با استفاده از یک مخزن EREW که حاوی گره‌هایی است که باید مطلع شوند و مسیرها نیز در آن علامت‌گذاری می‌شوند، خاصیت مستقل از دور بودن الگوریتم حفظ می‌شود.

فاز دوم: در بخش اول این فاز برگ‌های سالم درخت سطری اول به ریشه‌های دیگر درخت های سطری و در بخش دوم به ریشه‌های درخت‌های ستونی اطلاع‌رسانی می‌کنند. در بخش اول، در حالت خطادار بودن گره‌های توری، در شرایطی که حداقل یک برگ سالم در هر درخت سطری به صورت ستون‌مجزا وجود داشته باشد، نیاز به افزایش دور نیست. اگر $N = 2^n$، در اینصورت از بین 2^{N^2} حالت مختلف سالم و خطادار بودن برگ‌ها، در $N!*2^{N^2-N}$ حالت، بدون ازدیاد دور می‌توان خطای حداکثر $N^2 - N$ برگ را تحمل نمود. در شرایطی غیر این، بصورت حریصانه عمل کرده، براساس تعداد برگ‌های سالم هر درخت سطری، گره‌ای از درخت سطری اول را جهت اطلاع‌رسانی به ریشه یک درخت سطری منتسب می‌کنیم؛ با انجام این کار در دورهای متوالی تمامی ریشه‌های درخت‌های سطری مطلع گردند. اگر تعداد برگ‌های خطادار درخت سطری اول d باشد، تعداد برگ‌های سالم این درخت $2^n - d$ خواهد بود. دورهای الگوریتم حداکثر به تعداد $\lceil 2^n/(2^n-d) \rceil$ اضافه خواهد شد. اگر فرض کنیم که برگ x برای اطلاع رسانی به ریشه RT_y منتسب شده، مسیر طی شده از CT_x، به سمت برگ $((RT_y,x),(CT_x,z))$ خواهد بود و از آنجا به سمت ریشه مقصد خواهد رفت که تداخلی با هیچ مسیر اطلاع‌رسانی دیگری ندارد. خطای گره سطح i-ام باعث تقسیم زیر شاخه آن به دو زیر شاخه در سطح i+1 می‌گردد. در بدترین حالت خرابی تمامی گره‌های یک درخت از سطوح 0 تا i-1 موجب تقسیم آن به 2^i زیردرخت مجزا می‌گردد، که به ریشه‌های هر کدام از زیردرخت‌ها به طور جداگانه اطلاع‌رسانی می‌شود. اگر گره مبدأ s با مرتبه r(s) در یک درخت سطری و گره مقصد d با مرتبه r(d) در درخت سطری دیگری قرار داشته باشند و $r(s) \le r(d)$، آنگاه حداقل $2^{r(d)}$ کوتاه‌ترین مسیر مختلف از s به d وجود دارد [۲] که فرض می‌کنیم از این تعداد حداقل $1/\alpha$ (α عدد صحیح مثبتی است) هم‌چنان قابل استفاده می‌باشند. به طور متوسط تعداد گره‌های آگاه‌شده در هر دور دو برابر می‌شود، که برای اطلاع‌رسانی از آنها در دور بعد کمک می‌گیریم. برای حفظ هم‌بندی شبکه حداقل nfr >0 ریشه سطری سالم باقی می‌مانند. در بدترین حالت در هر درخت سطری j به تعداد 2^{k_j} زیردرخت که ریشه‌های آنها سالم و در سطح k_j قرار دارند باید اطلاع‌رسانی کرد. در این حالت تعداد خطاها به میزان $O(2^{k_j})$ در هر درخت سطری خواهد بود. به این ترتیب که در مرحله نخست، در درخت‌های سطری فرد به اولین زیردرخت سمت راست، و در درخت‌های سطری زوج به اولین زیردرخت سمت چپ خبر می‌دهیم. در مرحله بعد گره اطلاع یافته در درخت فرد، به گره‌ای در اولین زیردرخت ایزوله شده بی‌اطلاع از سمت راست در درخت خود خبر داده و گره اطلاع یافته در درخت زوج عکس این کار را می‌کند. در مراحل بعد به صورت یک در میان زیردرخت های چپ و راست در درخت‌های سطری فرد و زوج مطلع می‌شوند. هر گره اطلاع‌یافته نیز در مرحله بعد این الگو را تکرار می‌نماید تا در نهایت تمامی زیردرخت‌های ایزوله شده در درخت‌های سطری حاوی یک گره

Algorithm BCast_MT(n, n, s_k) $s_k = (0^k, 0^n), 0 \le k \le n$ is the source of BCast

```
/* Phase 1 : Cover all leaves of first row – tree */
for r = 1 to n do
    for all x ∈ {0,1}^(r−1) do in parallel
        if (x is faulty)
            discard subtree rooted at x;
            faultyList ← x
        else
            if (x1 is nonFaulty)
                send the packet from covEredBy(x, s) to x1 through a route in this tree;
                where
                covEredBy(x, s) = { s                    if x = ε,
                                  { x                    if x = x'1,
                                  { covEredBy(x', s)     if x = x'0.
            else
                send the packet from covEredBy(x, s) to x10^k through a greedy route via other trees;
                where k is a number s.t. x10^k is the first left descendant node of x1 who is nonFaulty.
                In next steps every time x1 is responsible for transmitting a packet, x10^k transmits
                the packet instead.
for any covered – leave, x, do in parallel
    while (faultyList ≠ φ)
        take the first node of faultyList, y, and remove it from list;
        send packet from covEredBy(x, s) to y1 through a greedy – route via other trees;
```

```
/* Phase 2 : inform the roots of isolated sub trees of row – trees */
for all x ∈ {0,1}^n do in parallel
    if (x ≠ 0^n & there is no faulty node on the route from node x to
        { y, the leftmost isolated subtree root in RT(x);  if x is even
        { y, the rightmost isolated subtree root in RT(x); if x is odd )
        send the packet from node (covEredBy(x, 0^k), 0^n) ∈ V(RT(0^n)) to y via a greedy route;
        put the roots of other isolated sub trees of RT(x) in unInformed RowSubTree RootsList_x;
    else if (the route has any faulty node)
        put the roots of isolated sub trees of RT(x) in unInformed RowSubTree RootsList_x;
for all x, where x is any of covered leaves of RT(0^n) or an informed node in RT(i)
    if (x ∈ RT(i))
        while (unInformed RowSubTree RootsList_i ≠ φ)
            y ← take the head of the unInformed RowSubTree RootsList_i and remove it;
            send the packet from x to y via a greedy route;
    else
        for i = 1 to 2^n
            while (unInformed RowSubTree RootsList_i ≠ φ)
                y ← take the head of the unInformed RowSubTree RootsList_i and remove it;
                send the packet from node (covEredBy(x, 0^k), 0^n) ∈ V(RT(0^n)) to y via a greedy route;
/* inform the roots of isolated sub trees of column – trees */
execute the above subroutine for column trees analogously
```

```
/* Phase 3 : inform partially tree nodes and completely mesh nodes */
apply phase 1 in all isolated row and column sub trees :
    { partial odd broadcast in odd subtrees                          /* analogous to phase 1 */
    {     partial even broadcast in even subtrees   /* exchange the role of right and left children in phase 1 */
```

```
/* Phase 4 : inform remaining tree nodes */
for all nodes x ∈ ⋃_{i=0}^{n−1} {0,1}^i ∧ x = x'1
    if (x is not faulty)
        send the packet from x to x'
for all nodes x ∈ ⋃_{i=0}^{n−1} {0,1}^i where x = x'{0,1} and x' is faulty
    if (x is not faulty)
```

شکل (۱): الگوریتم همه‌پخشی متحمل خطا در شبکه توری درخت‌ها

$$\text{for all } i = 1,2,\ldots,j;\ \text{where } x0^{j} \text{ exists and is not faulty}$$
$$\text{send the packet from } x \text{ to } x0^{i}$$
$$\text{for all nodes } x \in \bigcup_{i=0}^{n-1}\{0,1\}^{i}$$
$$\text{if } \{x \text{ is not faulty } \wedge x1 \text{ is not faulty } \wedge x1 \text{ is not inf ormed }\}$$
$$\text{send the packet from } x \text{ to } x1$$

ادامه شکل (۱): الگوریتم همه‌پخشی متحمل خطا در شبکه توری درخت‌ها

مطلع باشند. پس در مراحل متوالی، 1، 2، ...، و حداکثر 2^{k_j-1} گره در هر درخت سطری در اطلاع‌رسانی شرکت می‌نمایند. با توجه به این که در درخت‌های زوج و فرد به ترتیب از سمت چپ و راست اطلاع‌رسانی می‌کنیم، تداخل در مرحله دوم که هدف اطلاع‌رسانی به نیمه دوم درخت است می‌تواند در گذر دو بسته از یک ریشه در مسیرهای مختلف به وجود آید. با توجه به این که حداقل nfr ریشه سالم داریم تنها $\left\lfloor 2^n / nfr \right\rfloor$ دور به الگوریتم اضافه می‌شود. اطلاع‌رسانی در دو درخت مجاور در دو نیمه مقابل به صورت مستقل صورت می‌پذیرد و به طور متوسط $1/\alpha$ ریشه‌های زیردرخت‌ها سالم هستند که دراین‌حالت در مراحل متوالی α، 2α، ...، $2^{k-1}\alpha$ دور به الگوریتم افزوده می‌شود. پس در حالتی که گره ریشه تا گره‌های سطح k_j -ام درخت سطری j-ام ساختار توری درخت‌ها، خطادار باشند تعداد دورهای الگوریتم حداکثر $O(\alpha 2^k) + \left\lfloor 2^n / nfr \right\rfloor + \left\lfloor 2^n /(2^n - d) \right\rfloor$ دور، $k = \max\{k_1, k_2, \ldots, k_n\}$، افزایش می‌یابد. در بخش دوم نیز رفتار الگوریتم در صورت وجود خطا مشابه توضیحات بخش اول خواهد بود.

فاز سوم: در این فاز تمام گره‌های ساختار مطلع یا پوشانده می‌شوند. در پایان فاز ۲ در هر زیردرخت مجزا در درخت‌های سطری و ستونی، یک گره مطلع است که مبدأ همه‌پخشی در زیردرخت خود می‌باشد. تمام عملیاتی که در فاز ۱ در درخت $RT(0^n)$ انجام شد در این فاز در تمامی درخت‌های سطری و ستونی به غیر از $RT(0^n)$ انجام می‌شود. اگر در همه‌پخشی فاز ۱، به جای اطلاع‌رسانی به گره‌های به فرمت $x1$ به گره‌هایی با فرمت $x0$ اطلاع‌رسانی کنیم، در نهایت تمامی برگ‌های زوج مطلع خواهند شد (همه‌پخشی زوج). در زیردرخت‌های سطری و نیز ستونی، بصورت یک در میان همه‌پخشی زوج و فرد انجام می‌دهیم و درنهایت تمامی برگ‌ها پوشانده خواهند شد. این فاز حداکثر به اندازه n طول می‌کشد. زیرا بیشترین طول اجرای الگوریتم در درختی است که به علت عدم وجود گره خطادار، زیردرخت مجزا ندارد.

فاز چهارم: پس از فاز ۳ گره‌هایی که مطلع نشده‌اند در ۳ دسته قرار دارند: گره‌های پدر که نقش مسیریاب برای اطلاع‌رسانی به فرزندان سمت راست خود را بر عهده داشتند، برخی گره‌های میانی که برگ‌های زوج (فرد) زیردرخت‌های مجزا محسوب می‌شوند و در نهایت گره‌هایی که تنها فرزندان چپ آنها سالم هستند.

اگر تمامی گره‌ها سالم بودند، با یک مرحله اطلاع‌رسانیِ فرزندان راست به پدرشان، همه گره‌ها مطلع می‌شدند. با وجود گره‌های خطادار، ابتدا همه فرزندان راست به پدرشان خبر می‌دهند. در قدم بعد تمام گره‌هایی که پدرشان خطادار است به ترتیب به فرزندان چپ نامطلع خود خبر می‌دهند. نهایتا تمامی گره‌ها فرزند راست خود را مطلع می‌کنند. قسمت دوم، باید آن‌قدر تکرار شود تا تمامی فرزندان چپ مطلع باشند. این تکرار حداکثر به اندازه n طول می‌کشد (اگر حداقل n-1 گره در درخت خطادار باشند). پس اگر تعداد گره‌های خطادار برابر f باشد، دورهای افزوده شده هم‌چنان O(f) خواهد بود. این اطلاع‌رسانی‌ها تداخلی باهم ندارند.

۴- ارزیابی نهایی

در صورت سالم بودن تمامی گره‌ها، الگوریتم بالا در زمان بهینه 2n+2 (n ارتفاع ساختار توری درخت‌ها می‌باشد) انجام می‌شود. اگر f تعداد متوسط گره‌های خطادار در هر درخت، k سطح گره‌های خطادار، nfr تعداد ریشه‌های خطادار در توری درخت‌ها، $\alpha > 0$ ضریبی ثابت و کوچک، و d تعداد برگ‌های خطادار درخت مبدا همه‌پخشی باشد، زمان اجرای کلی الگوریتم به صورت زیر است: n+O(fk) + $O(\alpha f) + \left\lfloor 2^n / nfr \right\rfloor + \left\lfloor 2^n /(2^n - d) \right\rfloor$+n+O(f)، که در نهایت به 2n + O(fk) + $O(\alpha f)$ تبدیل می‌شود.

۵- نتیجه‌گیری

در این مقاله اثر خطا بر الگوریتم همه‌پخشی در شبکه توری درخت‌ها، تحت مدل ارتباطی لینک-یک‌طرفه و تک‌پورت با مسیریابی غیرحساس به مسافت را بررسی کردیم. با استفاده از روش تعبیه یک الگوریتم همه‌پخشی متحمل خطا ارائه نموده، نشان دادیم که وجود خطا در گره‌های شبکه باعث افزایش تعداد دورهای الگوریتم به‌طور خطی، با ضریب کوچک، نسبت به تعداد خطاها و سطح گره‌های خطادار می‌گردد. بهبود این مقدار جای کار دارد. با توجه به کار انجام‌شده می‌توان گفت توری درخت‌ها در حضور خطا نیز عملکرد خوبی دارد.

۶- مراجع

[1] P .Salinger, P. Tvrdik. *Optimal broadcasting and gossiping in one-port meshes of trees with distance-insensitive routing*. Parallel Computing 28 (2002) 627–647.

[2] F. Leighton, B.M. Maggs, R. K.Sitaraman, *On the fault tolerance of some popular bounded-degree networks*. Siam J. Comput, Vol. 27, No. 5, pp. 1303–1333, (1998)

[3] J.D. Shih. *Fault-tolerant wormhole routing for hypercube networks*. Info. Processing Letters 86 (2003) 93–100.

[4] F.T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. (1992)

[5] F. Leighton, B. Maggs, S. Rao, *Packet routing and job-shop scheduling in O(congestion+dilation) steps*. (1994)