Packet Routing in Networks with Long Wires

Ronald I. Greenberg
H.-C. Oh

University of Maryland
College Park
Wire Delay

- A popular assumption: unit wire delay
- Less and less realistic as parallel computers get large
- Let's solve fundamental problem of routing data in a parallel machine while accounting for wire delays.
Graph Model

- Before routing, packets in initial queues at nodes (processors) where generated.
- Packet can traverse edge and enter queue at end when edge queue not full.
- Time to traverse edge $e$.
Goals

• Minimize time to route
• Minimize required size of edge queues.

Depend on paths packets traverse, but we concentrate on scheduling packet movements once paths have been selected.
Relation to Job-Shop Scheduling

job = sequence of operations, each of specified duration on specified machine

job ↔ packet

machine ↔ edge

packet routing like job-shop scheduling with all operations on any given machine being of same duration
Lower Bounds on Time

- dilation $d = \max.$ no. of edges traversed by a packet
- generalized dilation $D = \max.$ over packet paths of sum of edge delays
- congestion $c = \max.$ over edges of no. of packets that cross
- generalized congestion $C = \max.$ over edges of \#packets $\times$ edge delay
Previous Work

- Leighton, Maggs, Rao — on-line w. unit delay:
  \[ O(c + d \lg(Nd)) \] time
  \[ O(\lg(Nd)) \] queue size
  \( N = \# \text{packets} \)

- Shmoys, Stein, Wein from job-shop scheduling — off-line but poly-time alg:
  \[ O((C + D) \frac{\lg^2(Nd)}{\lg \lg(Nd)}) \] time

- (high probability)
Our On-Line Results

- $O((C+D) \frac{\lg(Nd)}{\lg\lg(Nd)})$ time
- $O\left(\frac{\lg(Nd)}{\lg\lg(Nd)}\right)$ queue size (high probability)
- Cuts off a $\lg(Nd)$ from SSW
- Better than multiplying LMR by max. edge delay $d_{\text{max}}$: $O(cd_{\text{max}} + dd_{\text{max}} \lg(Nd))$
Refinements

Instead of \((C+D)\frac{\log(Nd)}{\log\log(Nd)}\):

- \(C \log^\varepsilon(Nd) + D \log(Nd)\), \(\varepsilon > 0\)
- \(C + D \log(Nd)\)
Non-Constructive Results

- LMR- unit delay: Every schedule of length \( O(c+d) \) with constant size queues (when paths are edge-simple)

- \( U_5(\frac{\log(d_{\text{max}})}{\log\log(d_{\text{max}})}) \) with \( O(d_{\text{max}}) \) size queues
On-Line Approach

1. Initially assume $d_{\text{max}}$ poly. in $N$ & $d$.

2. Produce "unconstrained" schedule: several packets on an edge at same time.

3. "Flatten" into legitimate schedule.
Unconstrained Schedule
Each packet waits in initial queue random no. of time steps from \([0, C] \Rightarrow\)
sched. length \(\leq C + D\).
Prob. of \(C\) packets on a given edge at time \(t\):
\[ p \leq \left( \frac{C}{2} \right)^2 \left( \frac{1}{C} \right)^2 \]
\[ \leq \left( \frac{eC}{2} \right)^2 \left( \frac{1}{C} \right)^2 \]
\[ = \left( \frac{e}{2} \right)^2 \]
Unconstrained cont.'d

Multiply \( p \leq \left( \frac{e}{Z} \right)^2 \) by no. of edges \((\leq Nd)\) and no. of time steps \((C+D)\).

Prob. still less than \( \frac{1}{(Nd)^k} \)

for \( Z = \Theta \left( \frac{\lg(Nd)}{\lg\lg(Nd)} \right) \).
Flattening

Unconstrained length $L_w \geq 8$ packets on an edge $\implies$ legitimate schedule of length $8L$. Route packets as soon as edge free except that edge traversal at time $t$ in unconstrained does not occur before $YT$ in legitimate sched.

- Pf. by induction
- Beats $SSW \times Ld_{\text{max}}$ by using fact that edge delays don't change.
Removing Restrictions

Don't really need $d_{\text{max poly}}$ in $N \& d$.  

Round edge delays down to nearest multiple of $\frac{d_{\text{max}}}{Nd}$. 

Work with time steps of length $\frac{d_{\text{max}}}{Nd}$. 

Then, to correct for roundoff, double clock period.
Off-Line Preliminaries

Use Lovász Local Lemma:

- "bad" events $A_1, A_2, \ldots, A_m$
- $\Pr\{A_i\} \leq p$
- dependence $\leq b$:
  each event mutually independent of at least $m-b$ others.

If $4pb < 1$, there is a non-zero prob. that no $A_i$ occurs.
Off-Line Approach

1. Successive refinement to "greedy schedule" (no waiting). Bound congestion in smaller and smaller intervals of time down to $\Theta(d_{max}^2)$.

2. Bound no. of packets using an edge in any unit time step.

3. Flatten as before.
Initial Refinement

Each packet gets delay from $[1, \alpha \cdot c \cdot d_{\text{max}}]$. WLOG $c \cdot d_{\text{max}} = D$, so schedule length $(1+\alpha) c \cdot d_{\text{max}}$.

Bad event for each edge:

$> \lg c$ packets on it during any interval of $d_{\text{max}}/\lg c$ time steps.

$p \leq (1+\alpha) c \cdot d_{\text{max}} (\lg c) \left( \frac{2 \cdot d_{\text{max}}/\lg c}{\alpha c \cdot d_{\text{max}}} \right) \lg c$

$b \leq c \cdot d = O(c^3)$

$4pb < 1$ for large enough constant $\alpha$. 
Sketch of Rest

- Maintain upper bnd. on congestion in interval at $O\left(\frac{1}{d_{\text{max}}}\right)$ and bring interval length down to $\Theta(d_{\text{max}}^2)$.
- From schedule with congestion $k = O(d_{\text{max}})$ and length $\Theta(kd_{\text{max}})$, produce $\Theta(kd_{\text{max}})$ length w. $O\left(\frac{\log d_{\text{max}}}{\log \log d_{\text{max}}}\right)$ packets on an edge at any time.
- Flatten.