An Efficient Counting Network

Costas Busch
Brown University

Marios Mavronicolas
University of Cyprus

We present a practical counting network which:

- Is efficient in terms of contention
- Its construction is based on new techniques

We proceed by presenting:

- Background for counting networks
- Our counting network construction
- Remarks and conclusions

Counting Networks

Introduced by Aspnes, Herlihy and Shavit in STOC 91

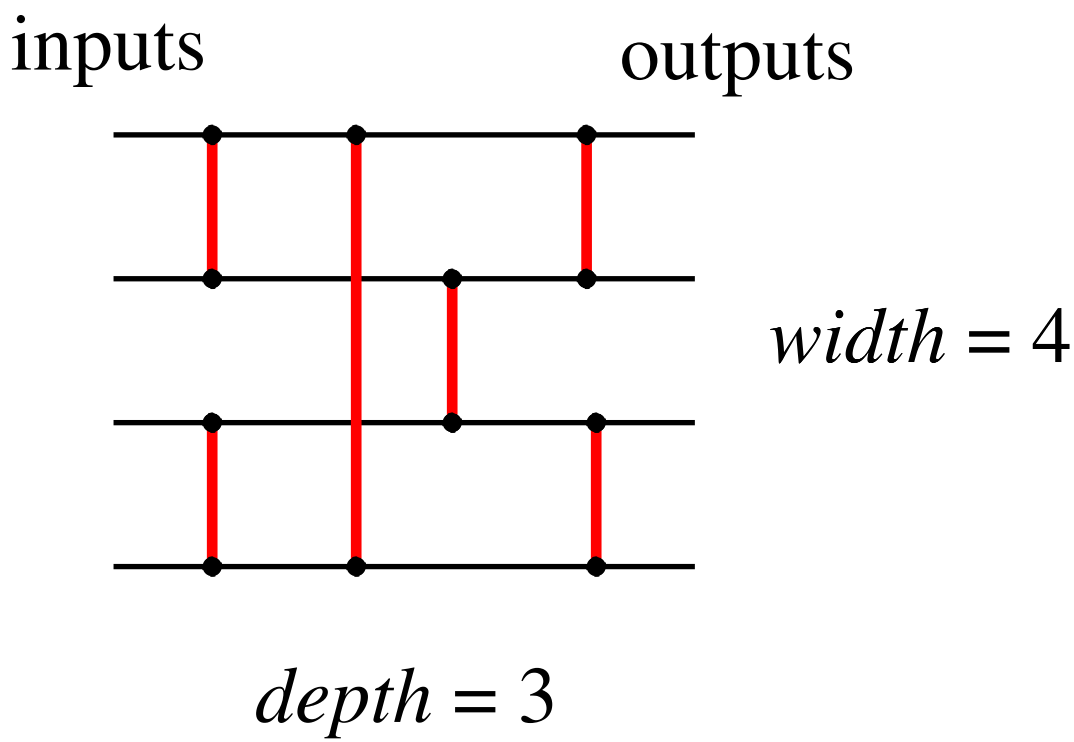
Distributed data structures used for:

- Shared counters
- Producer/consumer buffers
- Barrier synchronization

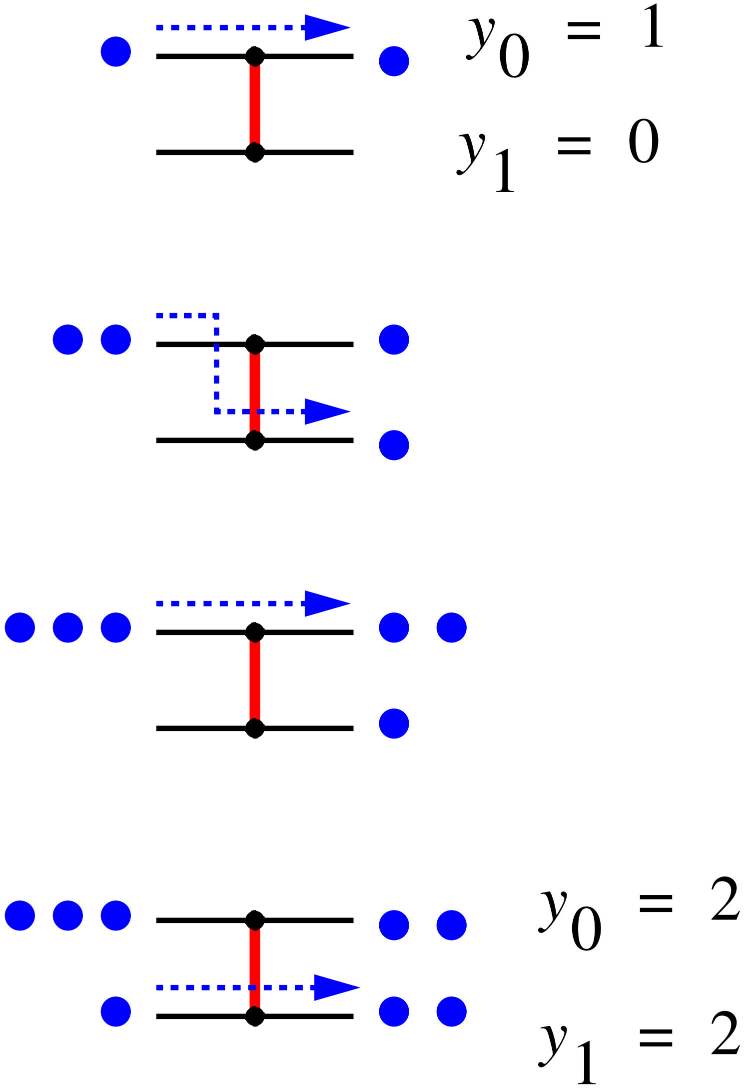
Advantages:

- Low contention
- Non-blocking

They look like Sorting Networks



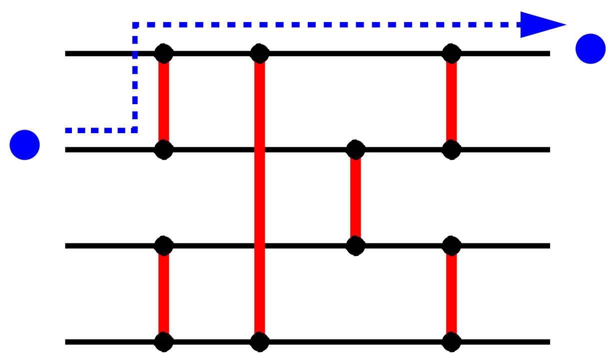
The ***Balancer***

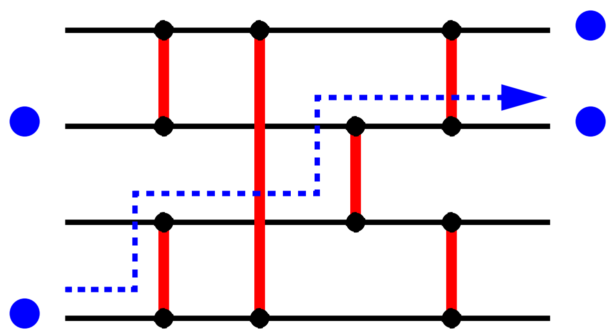


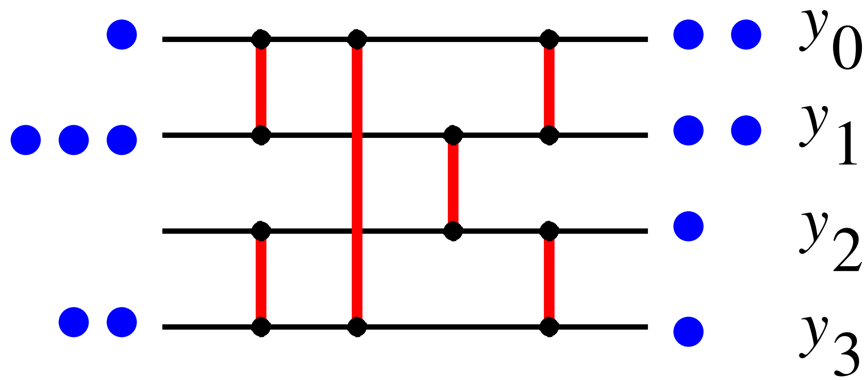
Step Property:

$$0 \le y_0 - y_1 \le 1$$

The counting network







Step Property:

$$0 \le y_i - y_j \le 1$$

for $\quad i < j$

Contention

- Concurrent processors access the same balancer at the same time

Amortized Contention **(Dwork et al. STOC 93)**

- The number of tokens goes to infinity

Practical counting networks:

- Bitonic
- Periodic

Have depth $O(\lg^2 t)$

Have amortized contention $O\left(\frac{n \lg^2 t}{t}\right)$

t = width

n = concurrency

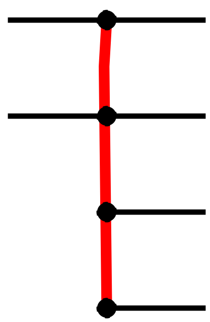
Problem:

- To decrease contention we increase depth

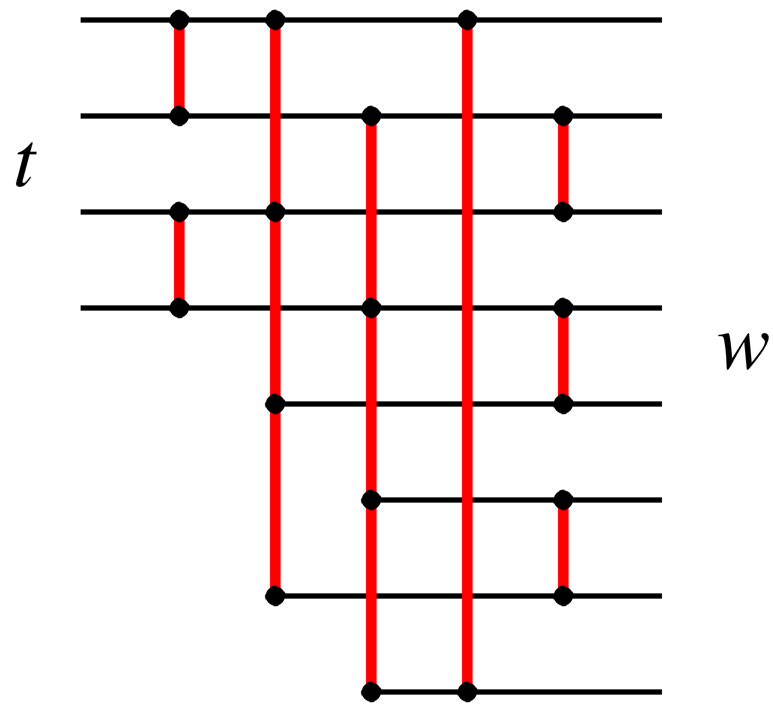
Our Counting Network

Has input width $t \leq w$ output width

Balancer



Counting Network



Has depth $O(\lg^2 t)$

Has amortized Contention $O\left(\frac{n\lg^2 t}{w} + \frac{n\lg t}{t}\right)$

t = input width

w = output width, $t \leq w$

n = concurrency

Advantages of increasing output width:

- Depth stays the same
- Contention decreases

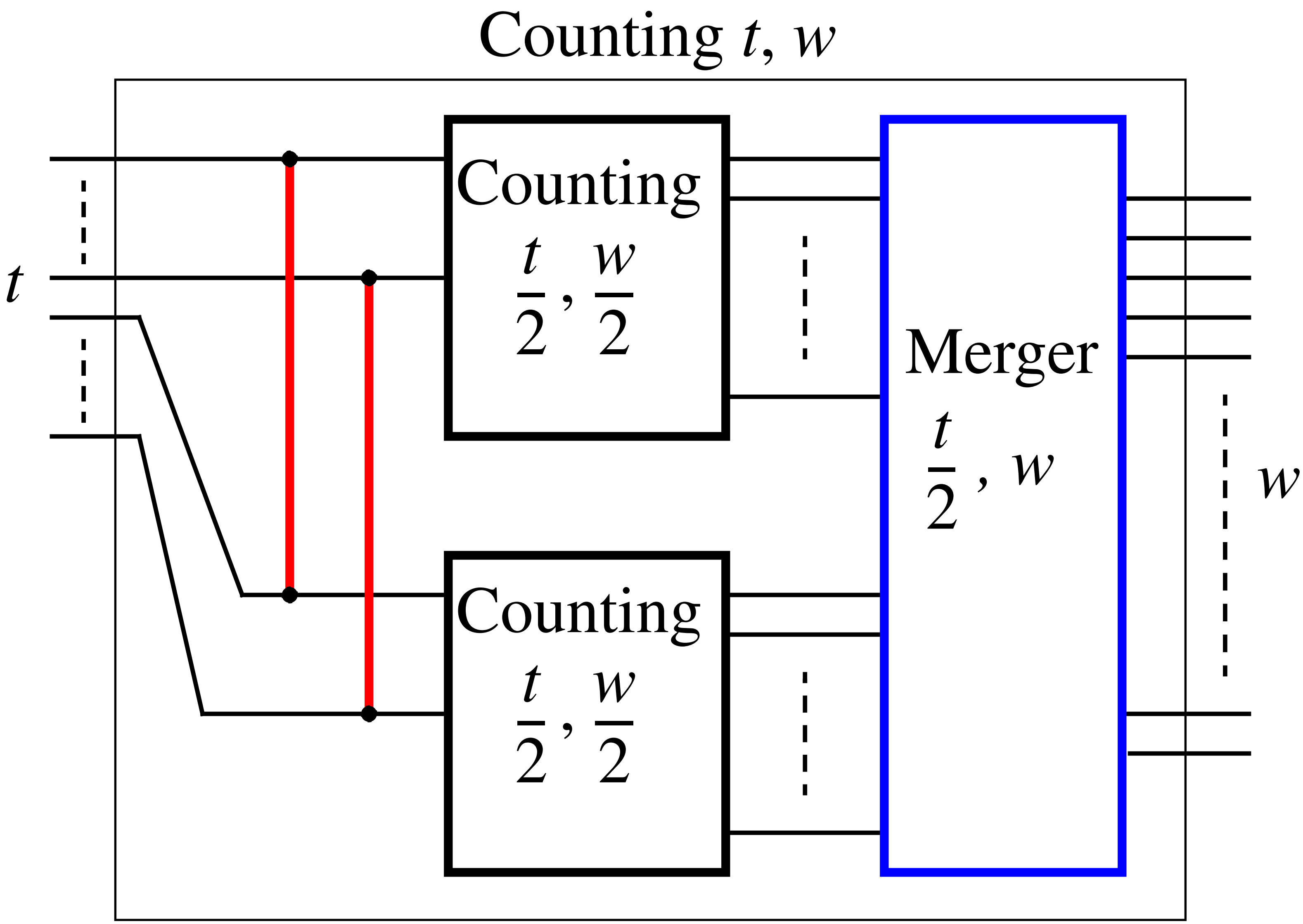
Settinng $w = O(t \lg t)$

we obtain contention $O\left(\frac{n \lg t}{t}\right)$

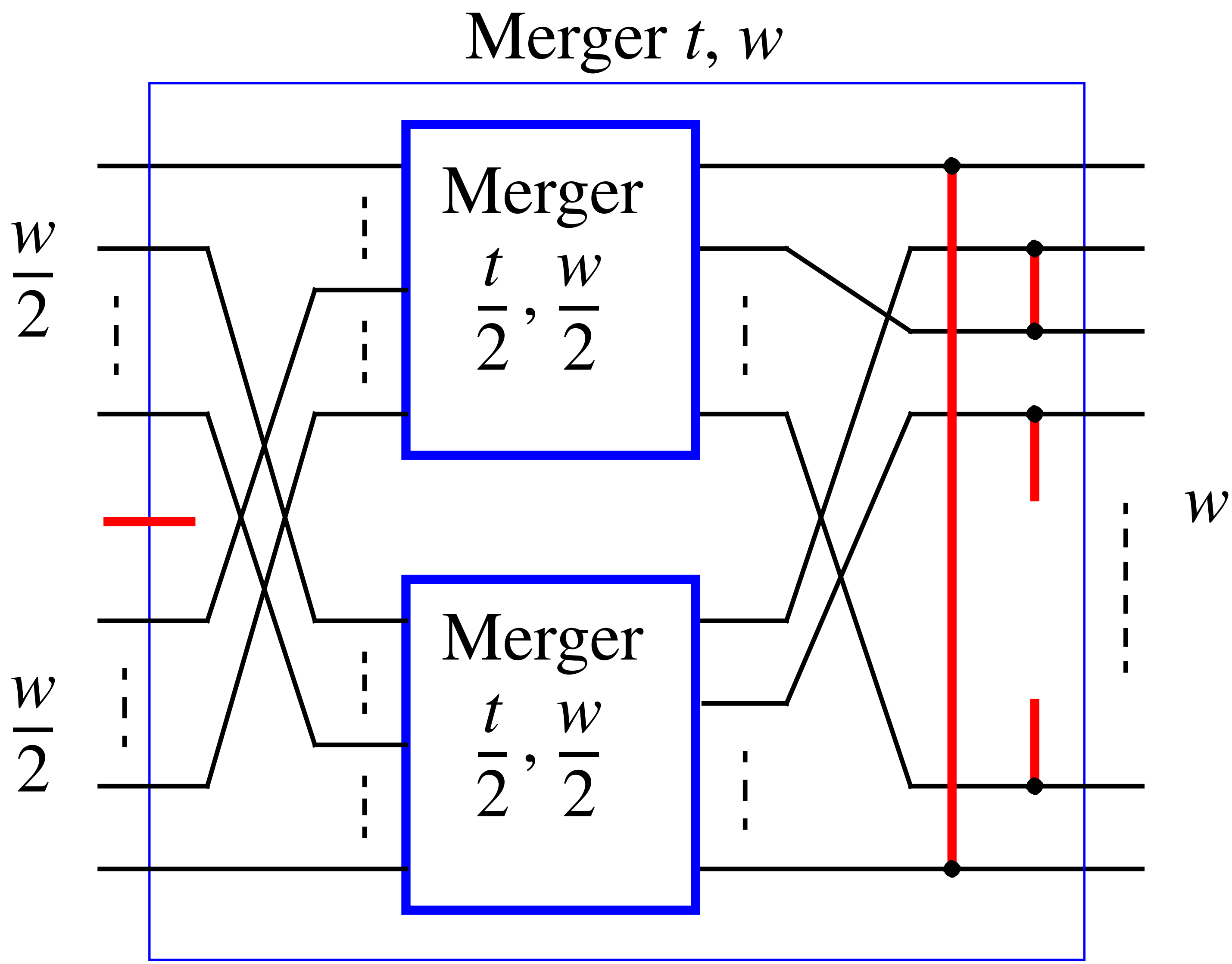
For input width t it achieves

- Same depth as other practical counting networks of width t
- Improves contention by a $\lg t$ factor

The construction



The *Bounded Difference Merger*



Remarks and Conclusions

Output width $p2^k$ (most have width 2^k)

Extend the construction to arbitrary widths

Use the Bounded Difference Merger for other constructions