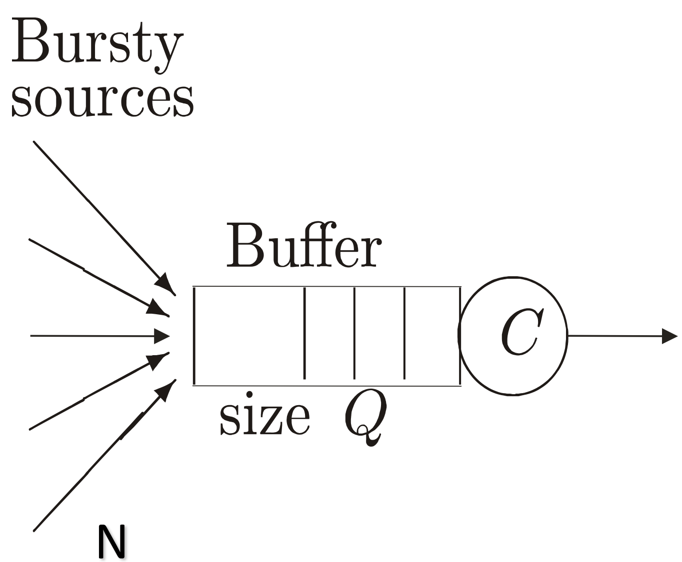
Parallel Simulation of ATM Switches Using Relaxation

A.S.McGough I.Mitrani

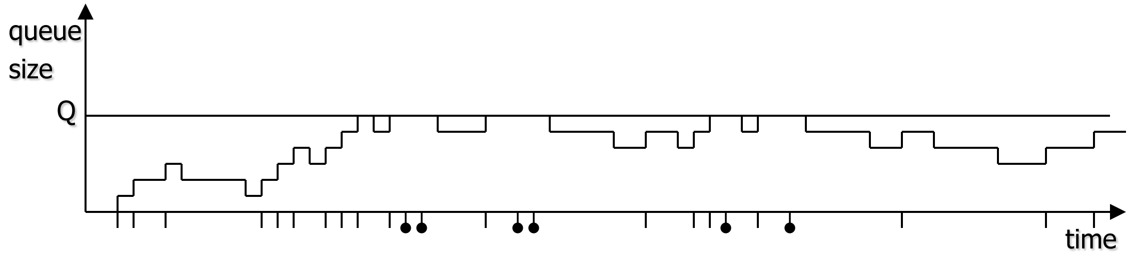
University Of Newcastle Upon Tyne

Model of an ATM switch

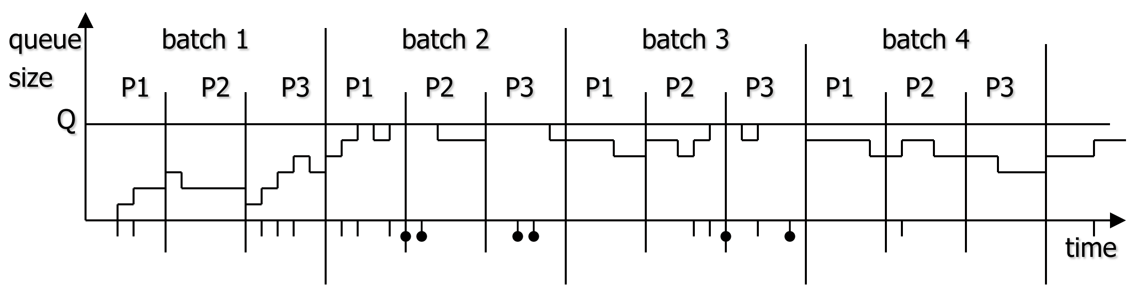
- Cells are generated by N independent bursty sources.
- There is an independent sequence of Off/On periods for each source.
- ATM server has a finite buffer of size Q, where cells are stored in order of arrival and the switch capacity is C cells per unit time.
- Cells finding a full buffer are lost.



- Simulation sample path:



- Performance measure:
  - Proportion of Cells lost: Lost Cells / Total Cells.
- Parallel algorithm:



- Recurrence equations and relaxation to resolve uncertainties.

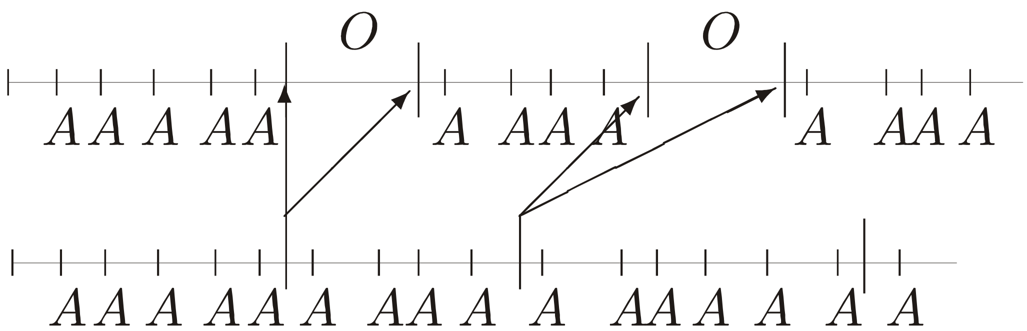
UNIVERSITY OF NEWCASTLE

Stages of The simulation

- Simulation proceeds in batches of B cells, each of which is processed in parallel by P processors.
  - Generate arrivals in parallel from each stream.
  - Merge arrivals.
  - Mark and remove lost cells. Two algorithms are presented, both requiring relaxation.
    - Algorithm 1 computes departure times.
    - Algorithm 2 computes buffer occupancy.

Generate arrivals in Parallel

- Generate the next B arrivals in each stream.



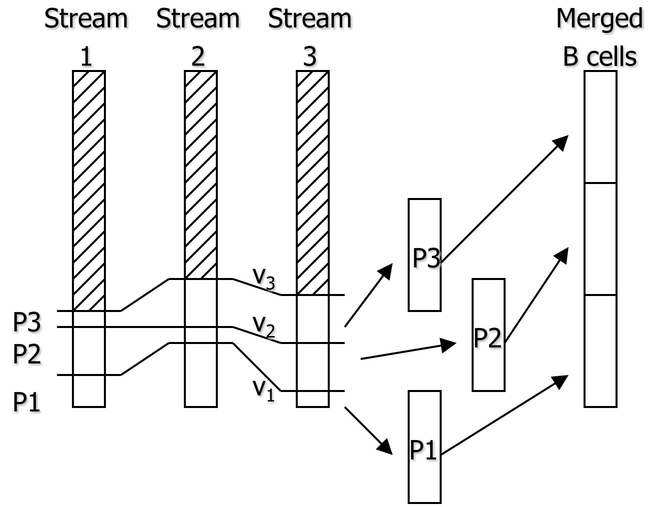
- Recurrence relation for arrivals:

$$A_{n+1} = A_n + \alpha_{n+1}$$

- Solve by parallel prefix.

Compute merged batch of arrivals

- Merge cells from the N arrival streams until we have B of them.
- Use a balanced parallel merge to ensure each processor does approximately equal work.
- All remaining cells are left for the next batch.



Mark and remove lost cells

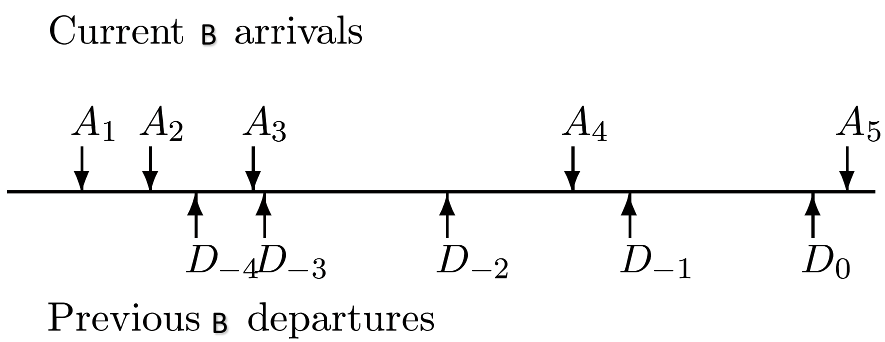
- Two algorithms are presented.

– Algorithm 1:
  - requires B=Q;
  - computes and stores the departure times.

– Algorithm 2:
  - works with arbitrary batch sizes.
  - computes the queue size seen by each arrival.
  - only state of queue after the last cell of a batch needs to be kept for the next batch.

- Both algorithms solve sets of recurrence relations in parallel, and use relaxation.

Algorithm 1



- Recurrence relation for departures:

$$D_{n+1} = \max(A_{n+1}, D_n) + c$$

- Cells 1 and 2 will be lost.
- Cell 5 will be accepted.
- Cells 3 and 4 may or may not be lost.
- Refine knowledge about uncertain cells by iteration.

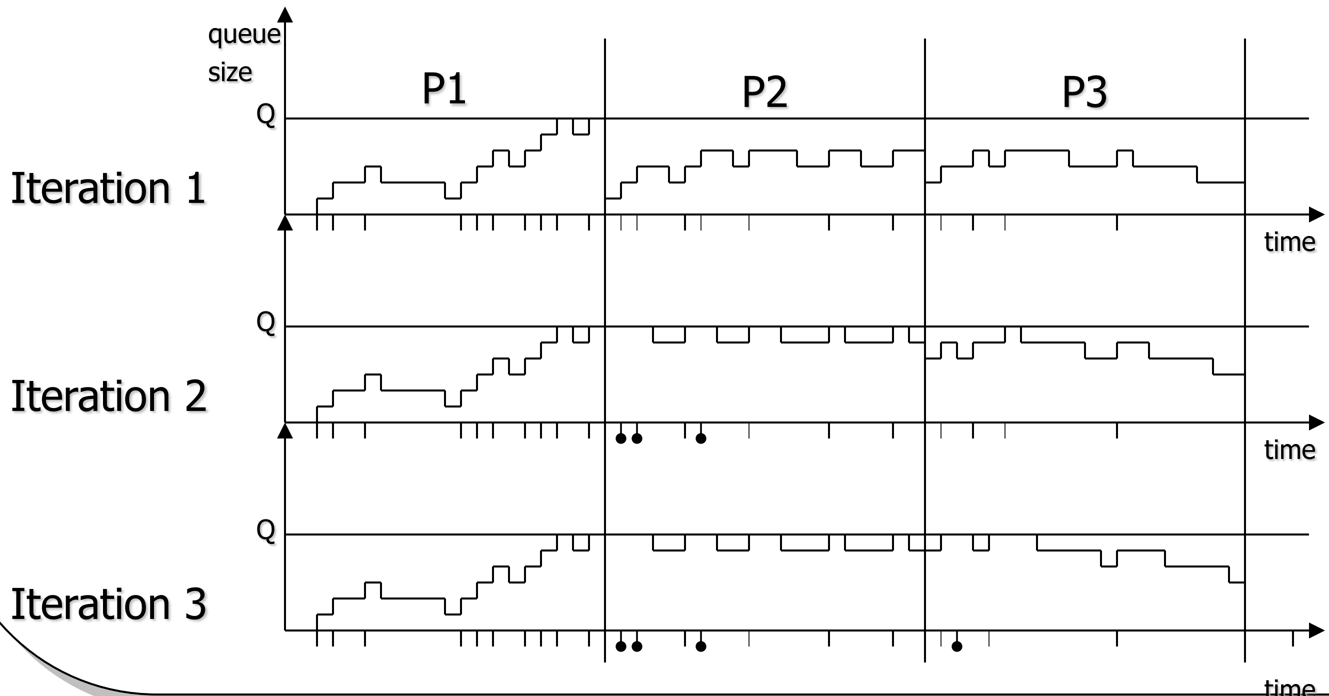
UNIVERSITY OF NEWCASTLE

Algorithm 2

- q_n is the queue size just before cell n arrives.
- d_n is the time of the last departure before cell n arrives or A_n if $q_n = 0$.
  - Solve the recurrences:

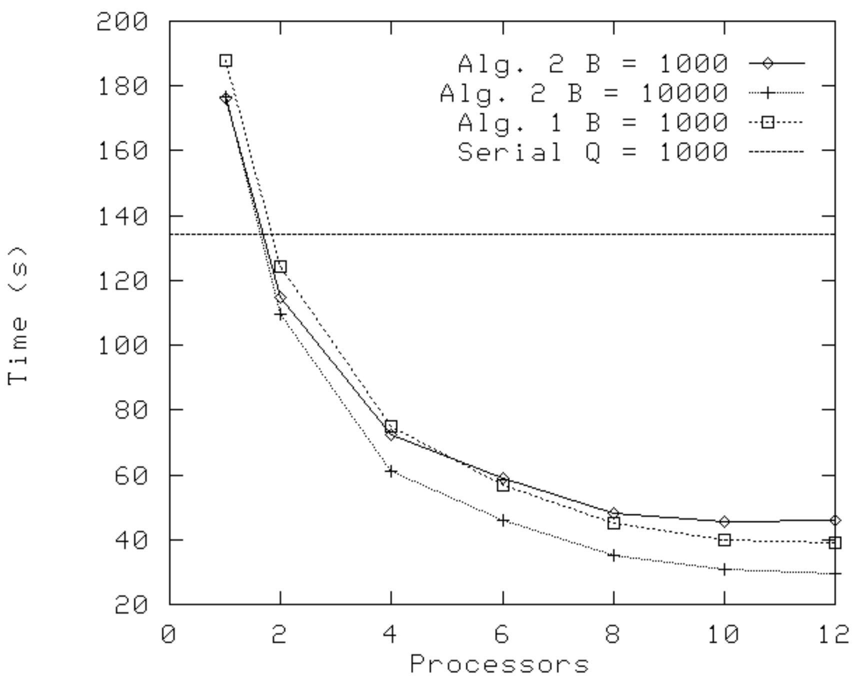
$$q_{n+1} = \max(q_n + \sigma_n - \delta_n, 0)$$

$$d_{n+1} = \begin{cases} d_n + \delta_n c & \text{if } \; q_{n+1} > 0 \\ A_{n+1} & \text{if } \; q_{n+1} = 0 \end{cases}$$

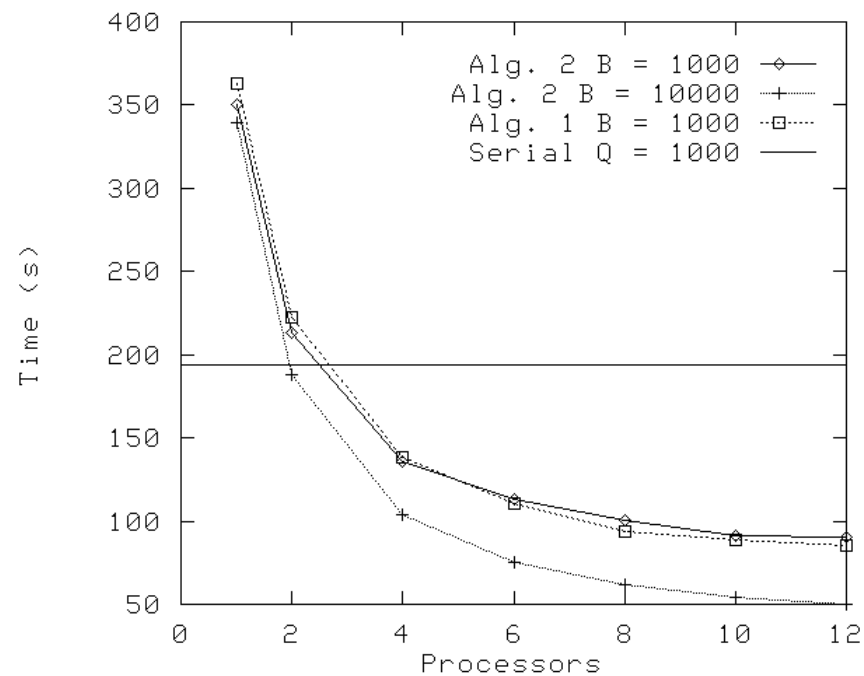


Results

- Graph of 6 stream inputs



- Graph of 24 stream inputs



Conclusion

- Speed-up obtained is almost linear.

$$T \sim O(M/P)$$

- This holds even in cases where cell loss is relatively high (1%).
- All random variables can have arbitrary distributions.
- There is a relationship between processor count and block size.
  - Each processor count has its own optimal block size.
- If Q is large then use algorithm 1 with B=Q.
- If Q is smaPll then use algorithm 2 with B larger than Q.