EE384y: Packet Switch Architectures

Matchings, implementation and heuristics

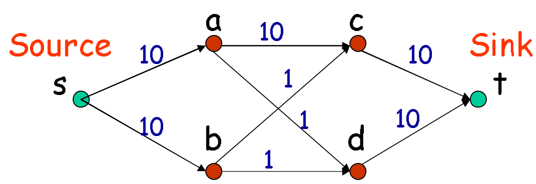
Nick McKeown
Professor of Electrical Engineering and Computer Science, Stanford University

nickm@stanford.edu
www.stanford.edu/~nickm

Outline

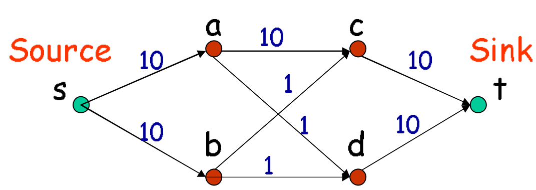
- Finding a maximum match.
  - Maximum *network flow* problems
    - Definitions and example
    - Augmenting paths
  - Maximum size/weight matchings as examples of maximum network flows
    - Maximum size matching
    - Complexity of maximum size matchings and maximum weight matchings
- What algorithms are used in practice?
  - Maximal Matches
  - Wavefront Arbiter (WFA)
  - Parallel Iterative Matching (PIM)
  - *i*SLIP

Network Flows

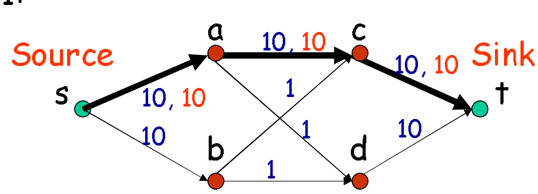


- Let $G = [V,E]$ be a directed graph with capacity $cap(v,w)$ on edge $[v,w]$.
- A *flow* is an (integer) function, f, that is chosen for each edge so that $f(v,w) \leq cap(v,w)$.
- We wish to maximize the flow allocation.

A maximum network flow example
By inspection



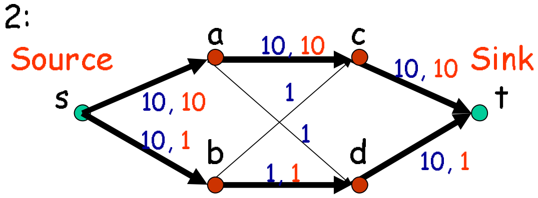
Step 1:



Flow is of size 10

A maximum network flow example

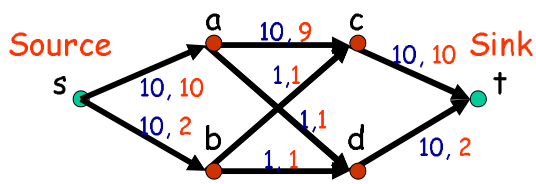
Step 2:



Flow is of size 10+1 = 11

Maximum flow:

Not obvious

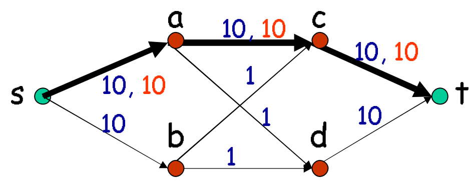


Flow is of size 10+2 = 12

Ford-Fulkerson method of augmenting paths

1. Set $f(v,w) = -f(w,v)$ on all edges.
2. Define a Residual Graph, $\boldsymbol{R}$, in which $res(v,w) = cap(v,w) - f(v,w)$
3. Find paths from s to t for which there is positive residue.
4. Increase the flow along the paths to augment them by the minimum residue along the path.
5. Keep augmenting paths until there are no more to augment.

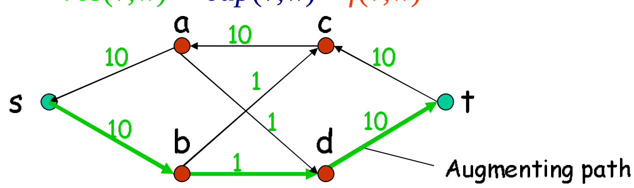
Example of Residual Graph



Flow is of size 10

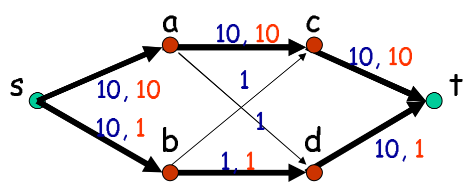
Residual Graph, R

$res(v,w) = cap(v,w) - f(v,w)$



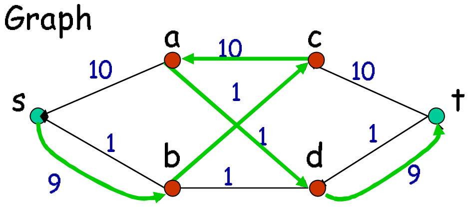
Example of Residual Graph

Step 2:



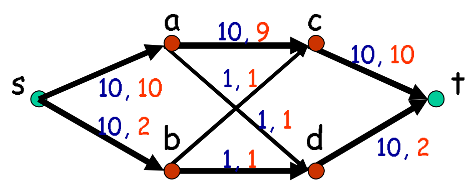
Flow is of size 10+1 = 11

Residual Graph



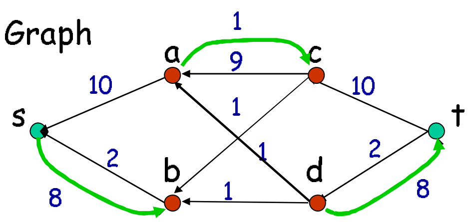
Example of Residual Graph

Step 3:



Flow is of size 10+2 = 12

Residual Graph



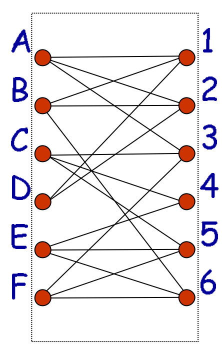
Complexity of network flow problems

- In general, it is possible to find a solution by considering at most $|V|.|E|$ paths, by picking *shortest augmenting path first.*
- There are many variations, such as picking *most augmenting path first.*

Outline

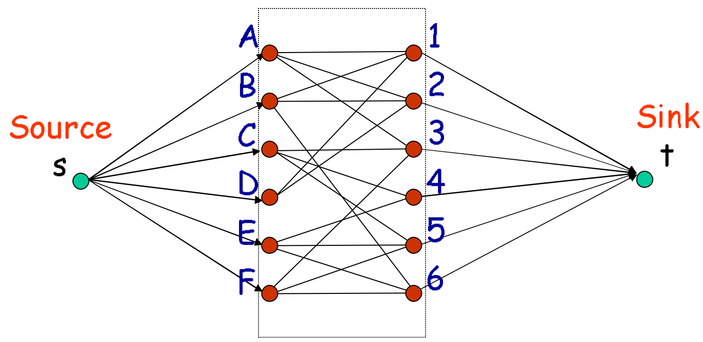
- Finding a maximum match.
  - Maximum *network flow* problems
    - Definitions and example
    - Augmenting paths
  - Maximum size/weight matchings as examples of maximum network flows
    - Maximum size matching
    - Complexity of maximum size matchings and maximum weight matchings
- What algorithms are used in practice?
  - Maximal Matches
  - Wavefront Arbiter (WFA)
  - Parallel Iterative Matching (PIM)
  - *i*SLIP

Finding a maximum size match



- How do we find the maximum size (weight) match?

Network flows and bipartite matching

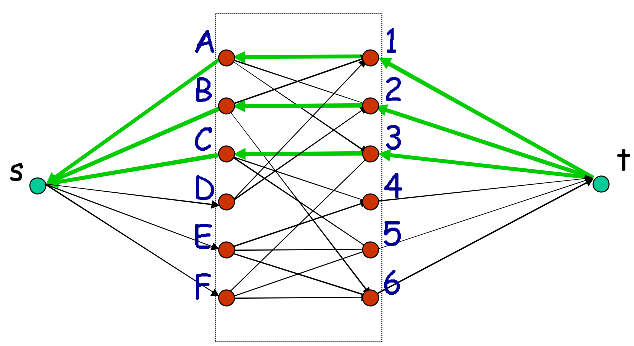


Finding a maximum size bipartite matching is equivalent to solving a network flow problem with capacities and flows of size 1.

Network flows and bipartite matching

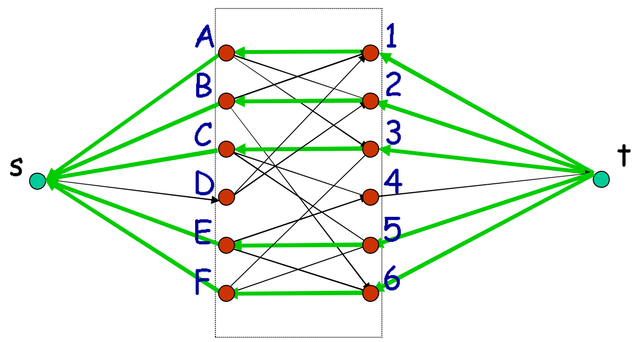
Ford-Fulkerson method

Residual Graph for first three paths:



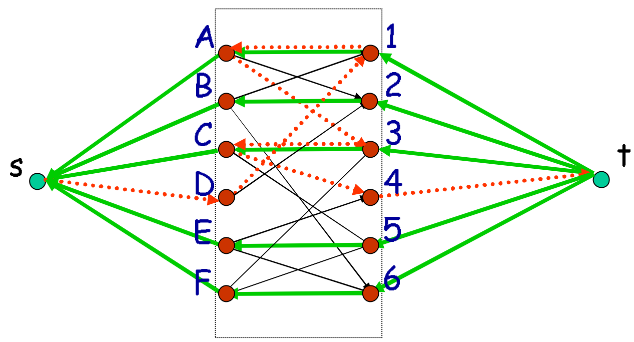
Network flows and bipartite matching

Residual Graph for next two paths:



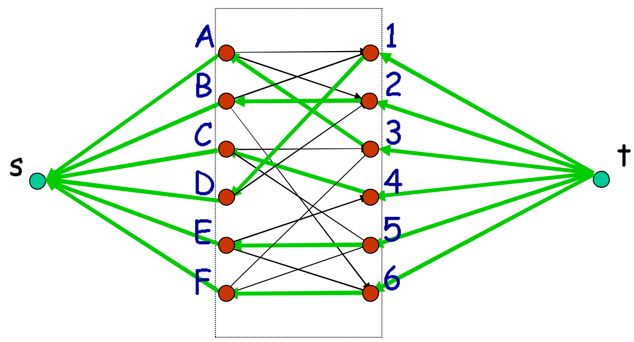
Network flows and bipartite matching

Residual Graph for augmenting path:



Network flows and bipartite matching

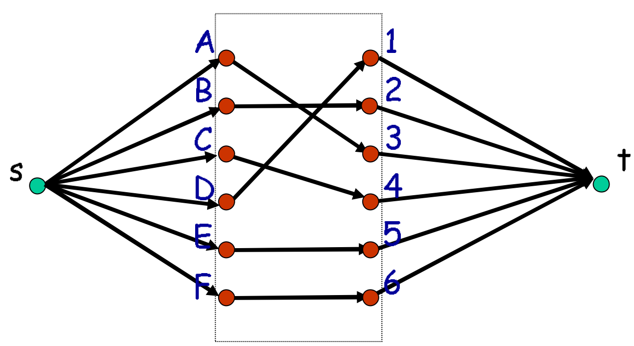
Residual Graph for last augmenting path:



Note that the path <u>augments</u> the match: no input and output is removed from the match during the augmenting step.

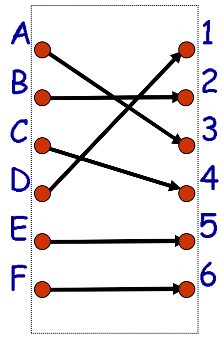
Network flows and bipartite matching

Maximum flow graph:



Network flows and bipartite matching

Maximum Size Matching:



Spring 2006

19

Complexity of Maximum Matchings

- Maximum Size Matchings:
  - Algorithm by Dinic $O(N^{5/2})$
- Maximum Weight Matchings
  - Algorithm by Kuhn $O(N^3)$
- In general:
  - Hard to implement in hardware
  - Slooooow.

Spring 2006

20

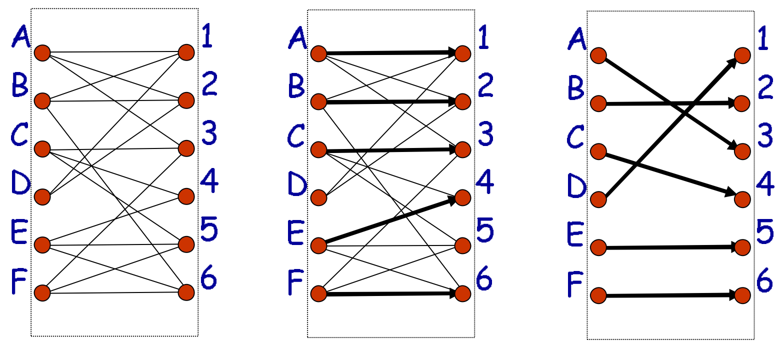
Outline

- Finding a maximum match.
  - Maximum *network flow* problems
    - Definitions and example
    - Augmenting paths
  - Maximum size/weight matchings as examples of maximum network flows
    - Maximum size matching
    - Complexity of maximum size matchings and maximum weight matchings
- What algorithms are used in practice?
  - Maximal Matches
  - Wavefront Arbiter (WFA)
  - Parallel Iterative Matching (PIM)
  - *i*SLIP

Maximal Matching

- A maximal matching is one in which each edge is added one at a time, and is not later removed from the matching.
- i.e. no augmenting paths allowed (they remove edges added earlier).
- No input and output are left unnecessarily idle.

Example of Maximal Size Matching



Maximal Size Matching

Maximum Size Matching

Maximal Matchings

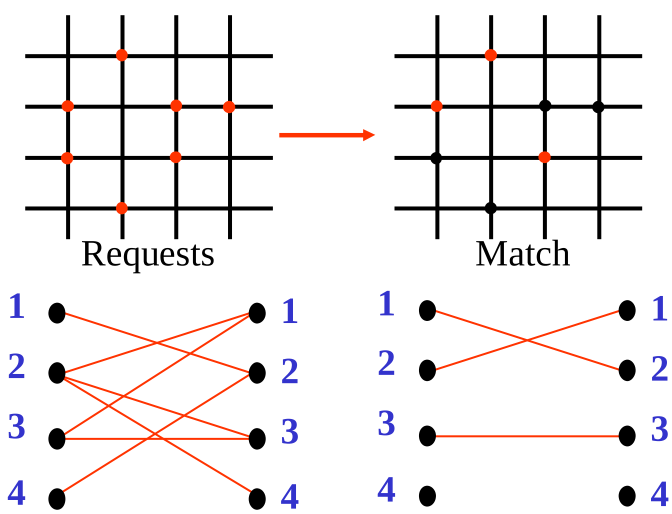
- In general, *maximal* matching is simpler to implement, and has a faster running time.
- A maximal size matching is at least half the size of a maximum size matching.
- A maximal weight matching is defined in the obvious way.
- A maximal weight matching is at least half the weight of a maximum weight matching.

Outline

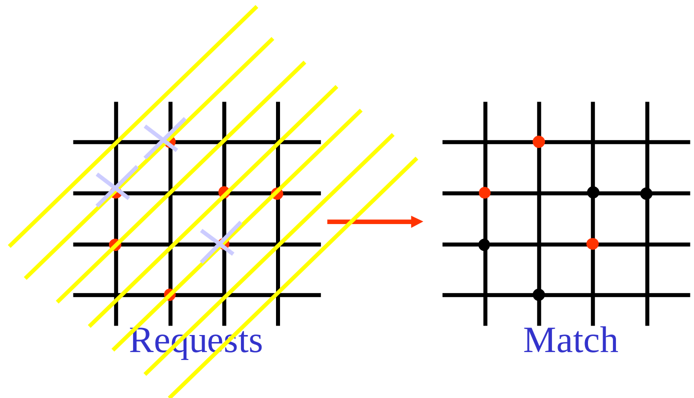
- Finding a maximum match.
  - Maximum *network flow* problems
    - Definitions and example
    - Augmenting paths
  - Maximum size/weight matchings as examples of maximum network flows
    - Maximum size matching
    - Complexity of maximum size matchings and maximum weight matchings
- What algorithms are used in practice?
  - Maximal Matches
  - Wavefront Arbiter (WFA)
  - Parallel Iterative Matching (PIM)
  - *i*SLIP

Wave Front Arbiter

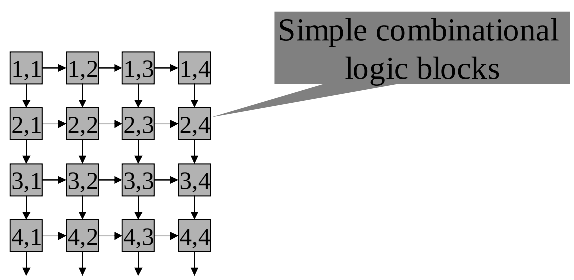
(Tamir)



Wave Front Arbiter

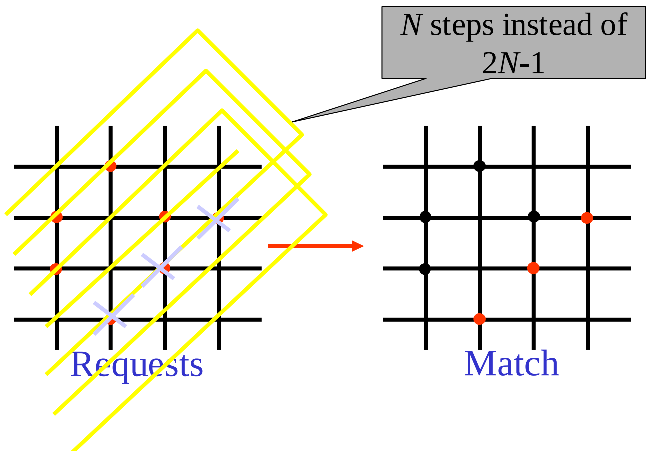


Wave Front Arbiter
Implementation



Wave Front Arbiter

Wrapped WFA (WWFA)



Wavefront Arbiters

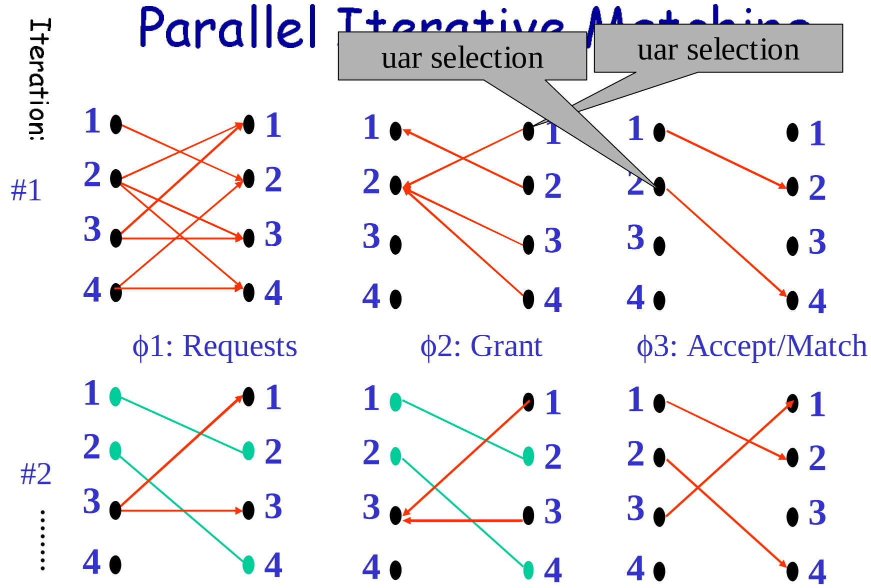
Properties

- Feed-forward (i.e. non-iterative) design lends itself to pipelining.
- Always finds maximal match.
- Usually requires mechanism to prevent Q_{11} from getting preferential service.
- In principle, can be distributed over multiple chips.

Outline

- Finding a maximum match.
  - Maximum *network flow* problems
    - Definitions and example
    - Augmenting paths
  - Maximum size/weight matchings as examples of maximum network flows
    - Maximum size matching
    - Complexity of maximum size matchings and maximum weight matchings
- What algorithms are used in practice?
  - Maximal Matches
  - Wavefront Arbiter (WFA)
  - Parallel Iterative Matching (PIM)
  - *i*SLIP

Parallel Iterative Matching



PIM Properties

- Guaranteed to find a maximal match in at most N iterations.
- In each phase, each input and output arbiter can make decisions independently.
- In general, will converge to a maximal match in $< N$ iterations.
- How many iterations should we run?

Parallel Iterative Matching

Convergence Time

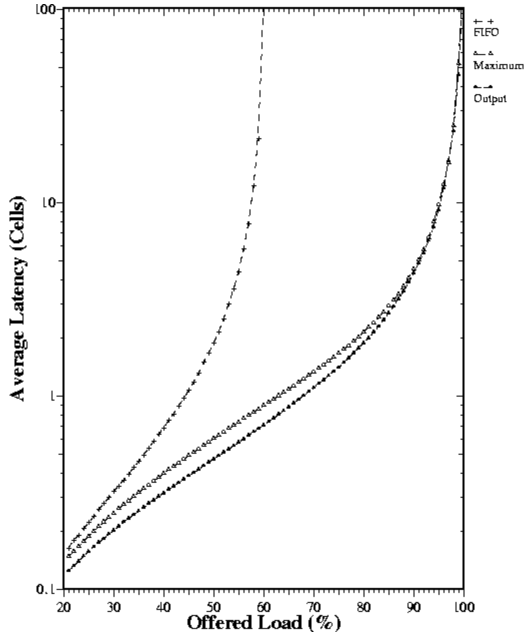
Number of iterations to converge:

$$E[U_i] \leq \frac{N^2}{4^i}$$

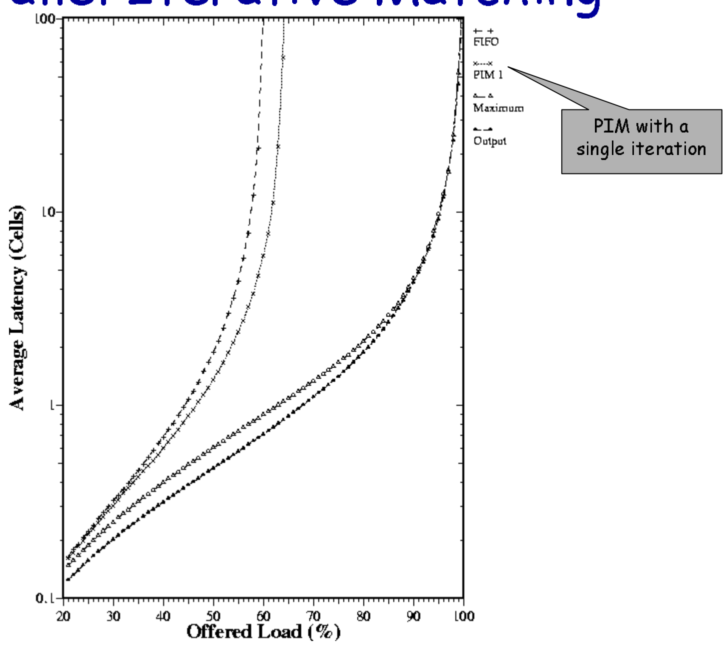
$$E[C] \approx \log N$$

C = # of iterations required to resolve connections
N = # of ports
U_i = # of unresolved connections after iteration i

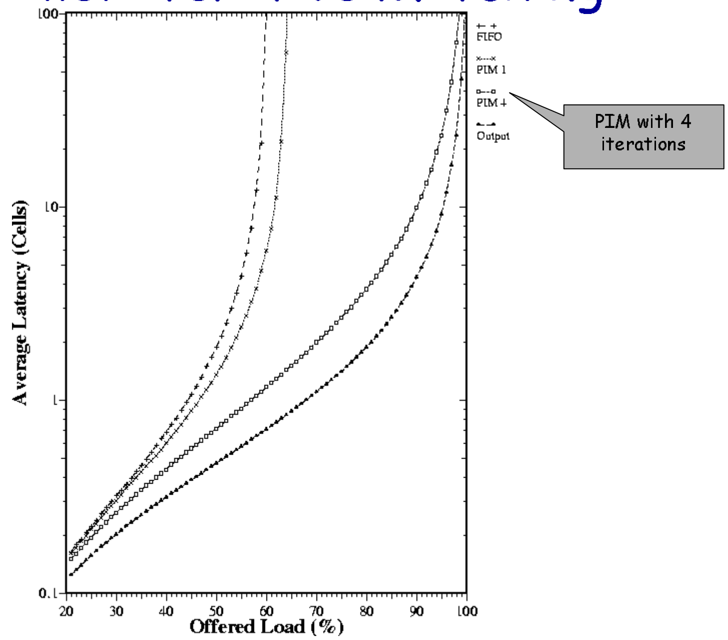
Parallel Iterative Matching



Parallel Iterative Matching



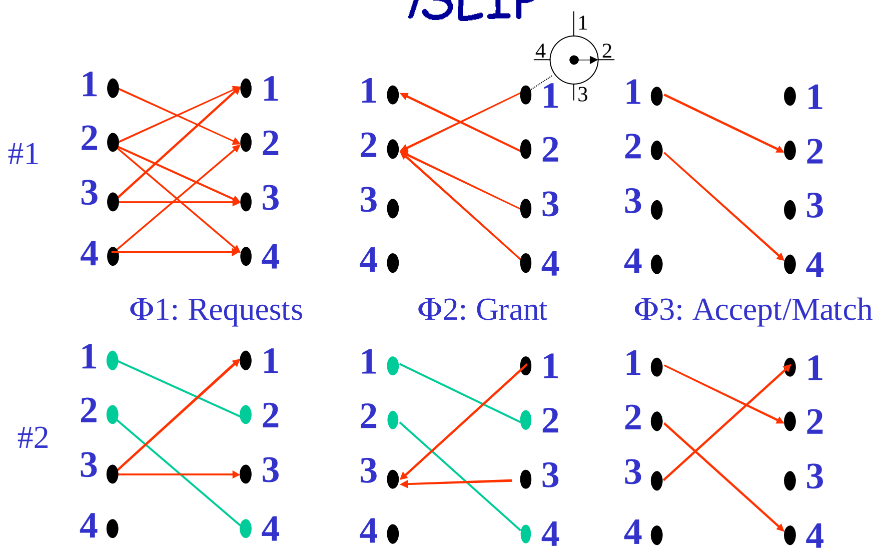
Parallel Iterative Matching



Outline

- Finding a maximum match.
  - Maximum *network flow* problems
    - Definitions and example
    - Augmenting paths
  - Maximum size/weight matchings as examples of maximum network flows
    - Maximum size matching
    - Complexity of maximum size matchings and maximum weight matchings
- What algorithms are used in practice?
  - Maximal Matches
  - Wavefront Arbiter (WFA)
  - Parallel Iterative Matching (PIM)
  - *i*SLIP

iSLIP



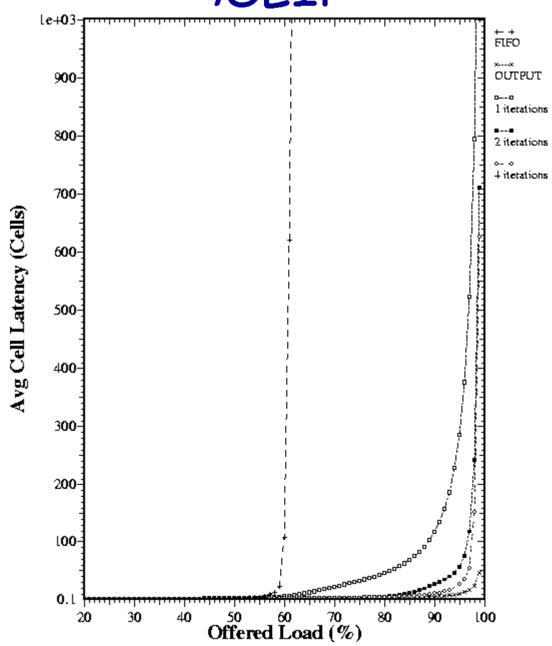
iSLIP Operation

- **Grant phase**: Each output selects the requesting input at the pointer, or the next input in round-robin order. *It only updates its pointer if the grant is accepted.*
- **Accept phase**: Each input selects the granting output at the pointer, or the next output in round-robin order.
- Consequence: Under high load, grant pointers tend to move to unique values.

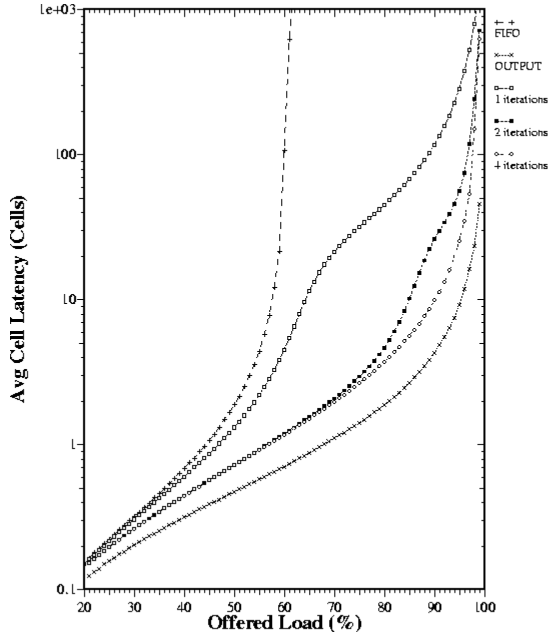
*i*SLIP
Properties

- Random under low load
- TDM under high load
- Lowest priority to MRU
- 1 iteration: fair to outputs
- Converges in at most N iterations. (On average, simulations suggest $< \log_2 N$)
- Implementation: N priority encoders
- 100% throughput for uniform i.i.d. traffic.
- But…some pathological patterns can lead to low throughput.

*i*SLIP

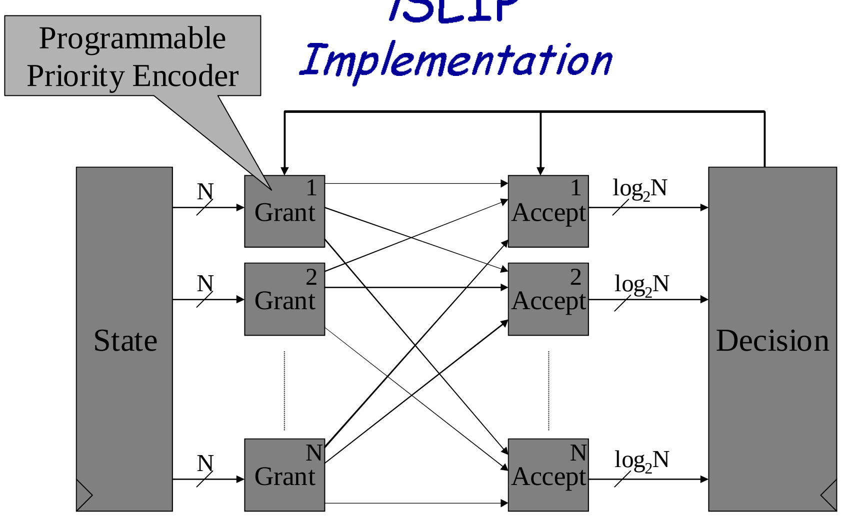


iSLIP



iSLIP

Implementation



Maximal Matches

- Maximal matching algorithms are widely used in industry (PIM, *i*SLIP, WFA and others).
- PIM and *i*SLIP are rarely run to completion (i.e. they are sub-maximal).
- A maximal match with a speedup of 2 is stable for non-uniform traffic.