Randomized Switch Scheduling Algorithms

Paolo Giaccone, Balaji Prabhakar and Devavrat Shah
Stanford University
Input queued switch

- Multiple queues and servers
• Crossbar constraints
  – each input can connect to at most one output
  – each output can connect to at most one input
Switch scheduling

Crossbar constraints
- each input can connect to at most one output
- each output can connect to at most one input
Switch scheduling

- Crossbar constraints
  - each input can connect to at most one output
  - each output can connect to at most one input
Switch scheduling

Crossbar fabric

• Crossbar constraints
  – each input can connect to at most one output
  – each output can connect to at most one input
Schedule = Bipartite graph matching
Scheduling algorithms

Practical Maximal Matchings

Gives < 100% Throughput

- iSLIP: Cisco’s GSR 12000 Series
  Nick McKeown

- Wave Front Arbiter
  Tamir and Chi

- Parallel Iterative Matching
  Anderson, Owicki, Saxe, Thacker
Scheduling algorithms

Practical Maximal Matchings
Gives < 100% Throughput

Max Size Matching
Gives < 100% Throughput
(McKeown–Ananthram–Walrand 96)

Max Wt Matching
Gives 100% Throughput!
(Tassiulas–Ephremides 92,
McKeown et. al. 96,
Dai–Prabhakar 00)
Switch algorithms

Maximal matching
< 100% Tput

Max Size Matching
< 100% Tput

Max Wt Matching
100% Throughput
Low queue buildup

Better performance

Easier to implement
Question

• Can we find an easy way to get the maximum weight matching?
  - Key idea: Randomization
Randomized approximation to the Max Wt Matching algorithm
Tassiulas’ algorithm

Previous schedule $S(t-1)$

Next time

Random Matching $R(t)$

Current schedule $S(t)$

MAX
Tassiulas’ algorithm

\[ S(t) \]

\[ W(S(t-1)) = 160 \]

\[ R(t) \]

\[ W(R(t)) = 150 \]
Performance of Tassiulas’ algorithm

**Theorem** (Tassiulas 98): The above scheme gives 100% throughput under any admissible Bernoulli IID inputs.
Backlogs under Tassiulas’ algorithm

![Graph showing mean IQ length vs. normalized load]

- **Tassiulas**
- **MVM**

Normalized Load

Mean IQ Length
Reducing backlogs: the Merge operation

$S(t-1)$
$W(S(t-1))=160$

$R(t)$
$W(R(t))=150$

Merge

30 v/s 120

130 v/s 30
Reducing backlogs: the Merge operation

\[ S(t-1) \]

\[ W(S(t-1)) = 160 \]

\[ R(t) \]

\[ W(R(t)) = 150 \]

\[ W(S(t)) = 250 \]
Performance of Merge algorithm

**Theorem** (Giaccone, Prabhakar and Shah): The Merge scheme gives 100% throughput under any admissible Bernoulli IID inputs.
Merge v/s Max

Normalized Load

Mean IQ Length

Tassiulas

Merge

MWM

Normalized Load
Use arrival information: Serena

S(t-1)

W(S(t-1))=209

The arrival graph
Use arrival information: Serena

\[ W(S(t-1)) = 209 \]

The arrival graph
Use arrival information: Serena

$S(t-1)$

$W(S(t-1)) = 209$

$S(t)$

$W(S(t)) = 243$

$W = 121$
Theorem (Giaccone, Prabhakar and Shah): The Serena algorithm gives 100% throughput under any admissible Bernoulli IID inputs.
Backlogs under Serena

Mean IQ Length vs Normalized Load

- Tassiulas
- Merge
- Serena
- MWM
Summary of main ideas and results

• Randomization alone is not enough

• Memory (using past sample) is very powerful

• Exploiting problem structure: the Merge operation

• Using arrival information