Optimal Oblivious Reconfigurable Networks

STOC 2022

Daniel Amir¹, Tegan Wilson¹, Vishal Shrivastav², Hakim Weatherspoon¹, Robert Kleinberg¹, Rachit Agarwal¹

¹Cornell University  ²Purdue University
How do we connect servers so they can communicate?
How do we connect servers so they can communicate?
How do we route messages along those connections?
Oblivious Reconfigurable Networks (ORNs)

• Set of $N$ nodes
Oblivious Reconfigurable Networks (ORNs)

- Set of $N$ nodes
- Edges reconfigure between each timestep according to a predefined schedule
Oblivious Reconfigurable Networks (ORNs)

- Set of $N$ nodes
- Edges reconfigure between each timestep according to a predefined schedule
- Route messages obliviously
Oblivious Reconfigurable Networks (ORNs)

• Set of $N$ nodes
• Edges reconfigure between each timestep according to a predefined schedule
• Route messages obliviously
  • Co-designing a connection schedule and routing protocol
Oblivious Reconfigurable Networks (ORNs)

• Set of $N$ nodes
• Edges reconfigure between each timestep according to a predefined schedule
• Route messages obliviously
  • Co-designing a connection schedule and routing protocol
• 1-regular directed networks for this talk
Oblivious Reconfigurable Networks (ORNs)

- Set of $N$ nodes
- Edges reconfigure between each timestep according to a predefined schedule
- Route messages obliviously
  - Co-designing a connection schedule and routing protocol
- 1-regular directed networks for this talk
  - Results extend to $d$-regular for any constant $d$
Oblivious Reconfigurable Networks
Oblivious Reconfigurable Networks

Diagram:

```
  a --> b
  |     |
  v     v
  d     c
  c     d
```

5
Oblivious Reconfigurable Networks
Oblivious Reconfigurable Networks
Oblivious Reconfigurable Networks
Oblivious Reconfigurable Networks

Virtual Topology
Oblivious Reconfigurable Networks

Virtual Topology
Oblivious Reconfigurable Networks

Virtual Topology
Oblivious Reconfigurable Networks

Virtual Topology

Route $a \rightarrow c$ starting at $t = 1$
Oblivious Reconfigurable Networks

Virtual Topology

Route $a \rightarrow c$ starting at $t = 1$
Oblivious Reconfigurable Networks

Virtual Topology

Route $a \rightarrow c$ starting at $t = 1$
Path has latency $L = 2$
Throughput
Throughput

Demand from $i \rightarrow j$ at timestep $t$
Throughput

• A matrix requests throughput $r$ if...

Demand from $i \to j$ at timestep $t$
Throughput

• A matrix requests throughput $r$ if...
  • Row/column sums $\leq r$

Demand from $i \rightarrow j$ at timestep $t$
Throughput

- A matrix requests throughput $r$ if...
  - Row/column sums $\leq r$

- An ORN design guarantees throughput $r$ if it can route all matrices requesting throughput $r$ without overloading edges

Demand from $i \rightarrow j$ at timestep $t$
Main Result

• Given a throughput value $r$, define an ORN design that:
Main Result

• Given a throughput value $r$, define an ORN design that:
  • Guarantees throughput $r$
Main Result

• Given a throughput value $r$, define an ORN design that:
  • Guarantees throughput $r$
  • Minimizes max latency $L$
Main Result

• Given a throughput value $r$, define an ORN design that:
  • Guarantees throughput $r$
  • Minimizes max latency $L$

• These objectives are in conflict with each other!
Main Result

• Given a throughput value \( r \), define an ORN design that:
  • Guarantees throughput \( r \)
  • Minimizes max latency \( L \)
• These objectives are in conflict with each other!

We fully resolve up to a constant factor!
Throughput v. Latency, $N = 10^{12}$
Throughput v. Latency, $N = 10^{12}$
Upper Bound
Upper Bound

![Graph showing the upper bound for Design 1]
Upper Bound

![Graph showing latency vs 1/throughput for Design 1 and Design 2. The graph illustrates the performance comparison between the two designs, with Design 1 generally showing lower latency across different throughput values.](image-url)
Lower Bound
Lower Bound
Lower Bound

![Graph showing latency vs. 1/throughput with two lines: one for uniform demands and one for general lower bound.](image)
Ongoing/Future Work

• ORN designs for all $N$ — not just infinitely many
• Semi-oblivious designs and analysis
  • Network still oblivious, but routing may be optimized for traffic
• Practical implementations — Daniel Amir
Thank You! Questions?

teganwilson@cs.cornell.edu