Power-aware Manhattan routing on chip multiprocessors

Anne Benoit[1], Rami Melhem[2],
Paul Renaud-Goud[1] and Yves Robert[1,3]

1. École Normale Supérieure de Lyon, France,
{Anne.Benoit — Paul.Renaud-Goud — Yves.Robert}@ens-lyon.fr

2. University of Pittsburgh, PA, USA, melhem@cs.pitt.edu

3. University of Tennessee Knoxville, TN, USA

Scheduling for Large Scale Systems Workshop
June 29, 2012

Motivation and introduction

Power-aware Manhattan routing
on chip multiprocessors

Motivation and introduction

Power-aware Manhattan routing
on chip multiprocessors

- Chip MultiProcessor (CMP): present and future of the processor

Motivation and introduction

Power-aware Manhattan routing
on chip multiprocessors

- Chip MultiProcessor (CMP): present and future of the processor
- Manhattan paths into a grid: good value for price

Motivation and introduction

Power-aware Manhattan routing
on chip multiprocessors

- Chip MultiProcessor (CMP): present and future of the processor
- Manhattan paths into a grid: good value for price
- Power issue crucial for both economical and environmental reasons

Motivation and introduction

Power-aware Manhattan routing
on chip multiprocessors

- Chip MultiProcessor (CMP): present and future of the processor
- Manhattan paths into a grid: good value for price
- Power issue crucial for both economical and environmental reasons
- Scalable links

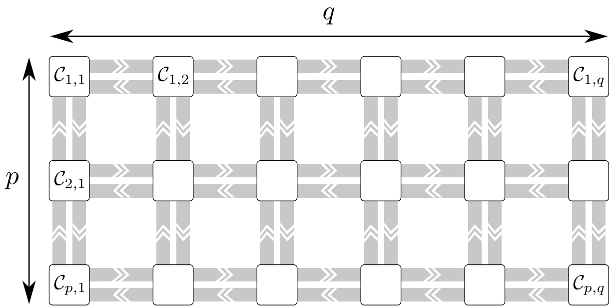
Outline of the talk

Outline of the talk

1. Framework
2. Theoretical results
3. Heuristics
4. Simulations

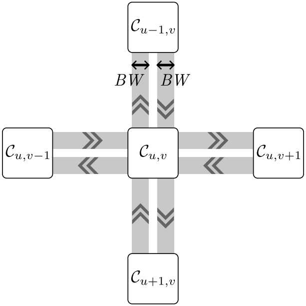
Platform, notations, and power consumption model

- Cores arranged onto a 2D grid



Platform, notations, and power consumption model

- Cores arranged onto a 2D grid
- Bi-directional links, but bandwidth not shared among two opposite directions



Platform, notations, and power consumption model

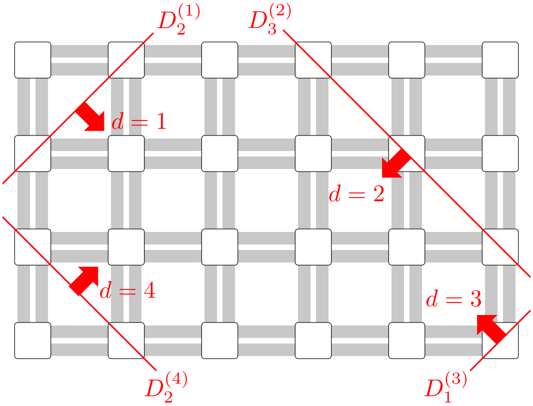
- Cores arranged onto a 2D grid
- Bi-directional links, but bandwidth not shared among two opposite directions
- $f_{(u,v)\rightarrow(u',v')}$: fraction of the bandwidth that is used

Platform, notations, and power consumption model

- Cores arranged onto a 2D grid
- Bi-directional links, but bandwidth not shared among two opposite directions
- $f_{(u,v)\to(u',v')}$: fraction of the bandwidth that is used
- $P_{\mathrm{dyn}}((u,v)\to(u',v')) = P_0 \times \left(f_{(u,v)\to(u',v')}BW\right)^{\alpha}$, where P_0 is a constant and $2 < \alpha \leq 3$
- $P_{(u,v)\to(u',v')} = P_{\mathrm{leak}} + P_0 \times \left(f_{(u,v)\to(u',v')}BW\right)^{\alpha}$. If $(u,v)\to(u',v')$ is inactive, then $P_{(u,v)\to(u',v')} = 0$.

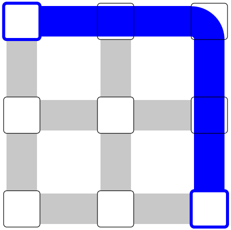
Communication model

- Communication defined by $\gamma_i = (\mathcal{C}_{usrc(i),vsrc(i)}, \mathcal{C}_{usnk(i),vsnk(i)}, \delta_i)$
- Direction d_i of communication γ_i
- Diagonal of cores $D_k^{(d)}$



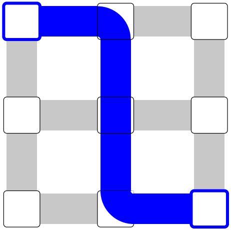
Routing definitions

- XY routing (XY):
  horizontally first, then vertically.



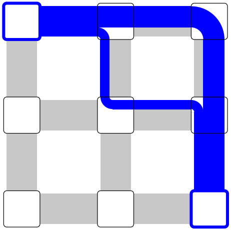
Routing definitions

- XY routing (XY):
  horizontally first, then vertically.
- Single-path Manhattan routing (1-MP):
  any shortest path



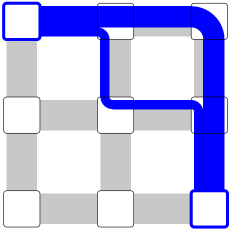
Routing definitions

- XY routing (XY):
  horizontally first, then vertically.
- Single-path Manhattan routing (1-MP):
  any shortest path
- s-paths Manhattan routing (s-MP):
  γ_i can be split into $s' \leq s$ distinct communications $\gamma_{i,1}, \gamma_{i,2}, \ldots, \gamma_{i,s'}$, of sizes $\delta_{i,1}, \delta_{i,2}, \ldots, \delta_{i,s'}$



Routing definitions

- XY routing (XY):
  horizontally first, then vertically.
- Single-path Manhattan routing (1-MP):
  any shortest path
- s-paths Manhattan routing (s-MP):
  γ_i can be split into $s' \leq s$ distinct communications $\gamma_{i,1}, \gamma_{i,2}, \ldots, \gamma_{i,s'}$, of sizes $\delta_{i,1}, \delta_{i,2}, \ldots, \delta_{i,s'}$
- max-paths Manhattan routing (max-MP):
  special case of s-MP where the number of paths is not bounded.
  (Remark: actually, there are $\binom{p+q-2}{p-1}$ Manhattan paths going from $\mathcal{C}_{1,1}$ to $\mathcal{C}_{p,q}$.)



Problem definition

We are given:

- a CMP
- a set of communications $\{\gamma_1, \ldots, \gamma_{n_c}\}$
- a routing rule (XY or s-MP), with a maximum number s of paths.

Problem definition

We are given:

- a CMP
- a set of communications $\{\gamma_1, \ldots, \gamma_{n_c}\}$
- a routing rule (XY or s-MP), with a maximum number s of paths.

Bandwidth must not be exceeded:
for all $(u, v) \in \{1, \ldots, p\} \times \{1, \ldots, q\}$ and $\mathcal{C}_{u',v'} \in succ_{u,v}$,

$$\sum_{\substack{i \in \{1,\ldots,n_c\}, j \in \{1,\ldots,s\} \\ (u,v)\to(u',v') \in path_{i,j}}} \delta_{i,j} \leq f_{(u,v)\to(u',v')} \times BW.$$

Problem definition

We are given:

- a CMP
- a set of communications $\{\gamma_1, \ldots, \gamma_{n_c}\}$
- a routing rule (XY or s-MP), with a maximum number s of paths.

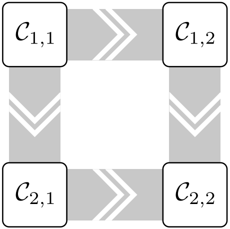
Bandwidth must not be exceeded:
for all $(u, v) \in \{1, \ldots, p\} \times \{1, \ldots, q\}$ and $\mathcal{C}_{u',v'} \in succ_{u,v}$,

$$\sum_{\substack{i \in \{1, \ldots, n_c\}, j \in \{1, \ldots, s\} \\ (u,v) \to (u',v') \in path_{i,j}}} \delta_{i,j} \leq f_{(u,v)\to(u',v')} \times BW.$$

$$\text{Minimize} \sum_{\substack{(u,v) \in \{1, \ldots, p\} \times \{1, \ldots, q\} \\ (u',v') \in succ_{(u,v)}}} P_{(u,v)\to(u',v')}$$

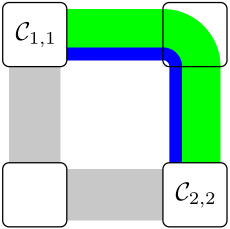
Quick comparison of routing rules

- $P_{\text{leak}} = 0$, $P_0 = 1$, $\alpha = 3$, $BW = 4$
- $\gamma_1 = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 1)$ and $\gamma_2 = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 3)$.



Quick comparison of routing rules

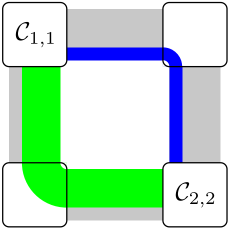
- $P_{\text{leak}} = 0$, $P_0 = 1$, $\alpha = 3$, $BW = 4$
- $\gamma_1 = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 1)$ and $\gamma_2 = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 3)$.



$$P_{\text{XY}} = 2 \times 4^3 = 128$$

Quick comparison of routing rules

- $P_{\text{leak}} = 0$, $P_0 = 1$, $\alpha = 3$, $BW = 4$
- $\gamma_1 = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 1)$ and $\gamma_2 = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 3)$.

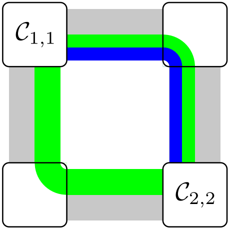


$P_{\text{XY}} = 128$

$P_{1-\text{MP}} = 2 \times (1^3 + 3^3) = 56$

Quick comparison of routing rules

- $P_{\text{leak}} = 0$, $P_0 = 1$, $\alpha = 3$, $BW = 4$
- $\gamma_1 = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 1)$ and $\gamma_2 = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 3)$.



$P_{\text{XY}} = 128$

$P_{1-\text{MP}} = 56$

$P_{2-\text{MP}} = 2 \times (2^3 + 2^3) = 32$

Outline of the talk

Manhattan vs XY; single source and destination

Theorem

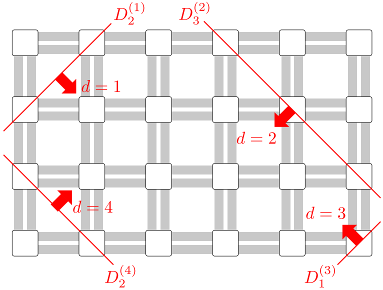
Given that $q = O(p)$, an upper bound of $P_{\mathrm{XY}}/P_{\max}$ is in $O(p)$.

Manhattan vs XY; single source and destination

Theorem

Given that $q = O(p)$, an upper bound of $P_{\mathrm{XY}}/P_{\max}$ is in $O(p)$.

- K: sum of all communications
- $K_k^{(1)}$: the sum of the γ_i that cross $D_k^{(1)}$
- In this case, $K_k^{(1)} = K$ for each k



Manhattan vs XY; single source and destination

Theorem

Given that $q = O(p)$, an upper bound of $P_{\mathrm{XY}}/P_{\max}$ is in $O(p)$.

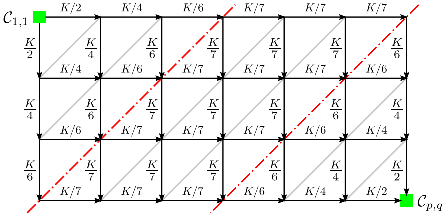
- K: sum of all communications
- $K_k^{(1)}$: the sum of the γ_i that cross $D_k^{(1)}$
- In this case, $K_k^{(1)} = K$ for each k
- $P_{\mathrm{XY}} = (p + q) \times K^{\alpha}$

Manhattan vs XY; single source and destination

Theorem

Given that $q = O(p)$, an upper bound of $P_{\mathrm{XY}}/P_{\max}$ is in $O(p)$.

- K: sum of all communications
- $K_k^{(1)}$: the sum of the γ_i that cross $D_k^{(1)}$
- In this case, $K_k^{(1)} = K$ for each k
- $P_{\mathrm{XY}} = (p+q) \times K^{\alpha}$
- Lower bound on $P_{\max}$. Ideal sharing of one communication:



Manhattan vs XY; single source and destination

Theorem

Given that $q = O(p)$, an upper bound of $P_{\mathrm{XY}}/P_{\max}$ is in $O(p)$.

$$P_{\max} \geq \sum_{k=1}^{p-1} 2k \left(\frac{K_k^{(1)}}{2k} \right)^{\alpha} + \sum_{k=p}^{q-1} (2p-1) \left(\frac{K_k^{(1)}}{2p-1} \right)^{\alpha} + \sum_{k=q}^{q+p-2} 2(q+p-k-1) \left(\frac{K_k^{(1)}}{2(q+p-k-1)} \right)^{\alpha},$$

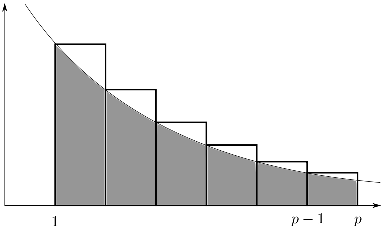
Manhattan vs XY; single source and destination

Theorem

Given that $q = O(p)$, an upper bound of $P_{\mathrm{XY}}/P_{\max}$ is in $O(p)$.

$$P_{\max} \geq \sum_{k=1}^{p-1} 2k \left(\frac{K_k^{(1)}}{2k} \right)^{\alpha} + \sum_{k=p}^{q-1} (2p-1) \left(\frac{K_k^{(1)}}{2p-1} \right)^{\alpha} + \sum_{k=q}^{q+p-2} 2(q+p-k-1) \left(\frac{K_k^{(1)}}{2(q+p-k-1)} \right)^{\alpha},$$

$K_k^{(1)} = K$ and $\sum_{k=1}^{p-1} k^{1-\alpha} \geq \int_1^p dx/x^{\alpha-1}$



Manhattan vs XY; single source and destination

Theorem

Given that $q = O(p)$, an upper bound of $P_{\mathrm{XY}}/P_{\max}$ is in $O(p)$.

$$P_{\max} \geq \sum_{k=1}^{p-1} 2k \left(\frac{K_k^{(1)}}{2k}\right)^{\alpha} + \sum_{k=p}^{q-1} (2p-1) \left(\frac{K_k^{(1)}}{2p-1}\right)^{\alpha} + \sum_{k=q}^{q+p-2} 2(q+p-k-1) \left(\frac{K_k^{(1)}}{2(q+p-k-1)}\right)^{\alpha},$$

$K_k^{(1)} = K$ and $\sum_{k=1}^{p-1} k^{1-\alpha} \geq \int_1^p dx/x^{\alpha-1}$, hence

$$P_{\max} \geq K^{\alpha} \left(2 \times \frac{1}{2^{\alpha-1}} \frac{1}{2-\alpha} \left(1 - p^{2-\alpha}\right) + \frac{q-p}{(2p-1)^{\alpha-1}}\right).$$

Manhattan vs XY; single source and destination

Theorem

Given that $q = O(p)$, an upper bound of $P_{\mathrm{XY}}/P_{\max}$ is in $O(p)$.

$$P_{\max} \geq \sum_{k=1}^{p-1} 2k \left(\frac{K_k^{(1)}}{2k}\right)^{\alpha} + \sum_{k=p}^{q-1} (2p-1) \left(\frac{K_k^{(1)}}{2p-1}\right)^{\alpha} + \sum_{k=q}^{q+p-2} 2(q+p-k-1) \left(\frac{K_k^{(1)}}{2(q+p-k-1)}\right)^{\alpha},$$

$K_k^{(1)} = K$ and $\sum_{k=1}^{p-1} k^{1-\alpha} \geq \int_1^p dx/x^{\alpha-1}$, hence

$$P_{\max} \geq K^{\alpha} \left(2 \times \frac{1}{2^{\alpha-1}} \frac{1}{2-\alpha} \left(1 - p^{2-\alpha}\right) + \frac{q-p}{(2p-1)^{\alpha-1}}\right).$$

Altogether, $P_{\max} = O(K^{\alpha})$ and $P_{\mathrm{XY}} = O(p \times K^{\alpha})$, hence the result.

Manhattan vs XY; single source and destination

Theorem

The upper bound of $P_{\mathrm{XY}}/P_{\max}$ in $O(p)$ is tight.

Manhattan vs XY; multiple sources and destinations

Theorem

Given that $q = O(p)$, an upper bound of $P_{\mathrm{XY}}/P_{\mathrm{max}}$ is in $O(p^{\alpha-1})$.

Theorem

The upper bound of $P_{\mathrm{XY}}/P_{\mathrm{max}}$ in $O(p^{\alpha-1})$ can be achieved with a $1\text{-}\mathrm{MP}$ routing on a square CMP.

NP-completeness of Manhattan routing

Theorem

Finding a s-MP routing that minimizes the total power consumption while ensuring that link bandwidths are not exceeded is a NP-complete problem.

Outline of the talk

Summary of the heuristics

- Simple greedy (**SG**): greedily assigns communications, hop by hop, on the least loaded link.

Summary of the heuristics

- Simple greedy (**SG**): greedily assigns communications, hop by hop, on the least loaded link.
- Improved greedy (**IG**): virtually pre-assigns communications onto links, then almost like **SG**.

Summary of the heuristics

- Simple greedy (**SG**): greedily assigns communications, hop by hop, on the least loaded link.
- Improved greedy (**IG**): virtually pre-assigns communications onto links, then almost like **SG**.
- Two-bend (**TB**): for each communication, chooses the best path with two bends.

Summary of the heuristics

- Simple greedy (**SG**): greedily assigns communications, hop by hop, on the least loaded link.
- Improved greedy (**IG**): virtually pre-assigns communications onto links, then almost like **SG**.
- Two-bend (**TB**): for each communication, chooses the best path with two bends.
- XY improver (**XYI**): starts from *XY* assignment, and moves communications from the highest loaded link.

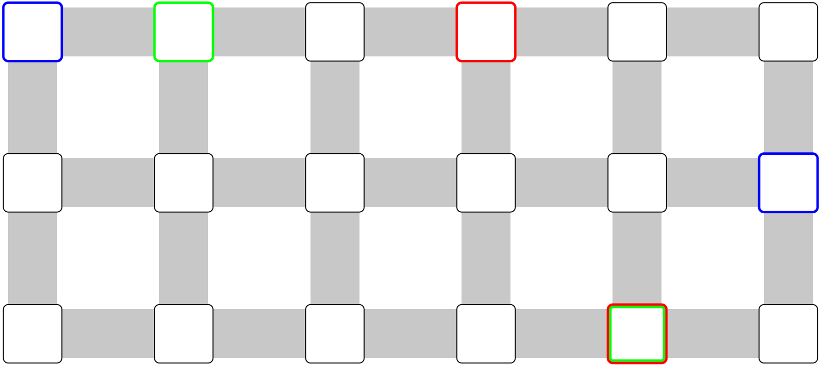
Summary of the heuristics

- Simple greedy (**SG**): greedily assigns communications, hop by hop, on the least loaded link.
- Improved greedy (**IG**): virtually pre-assigns communications onto links, then almost like **SG**.
- Two-bend (**TB**): for each communication, chooses the best path with two bends.
- XY improver (**XYI**): starts from *XY* assignment, and moves communications from the highest loaded link.
- Path remover (**PR**): virtually pre-assigns communications onto links, and iteratively prevents communications from using highly loaded links.

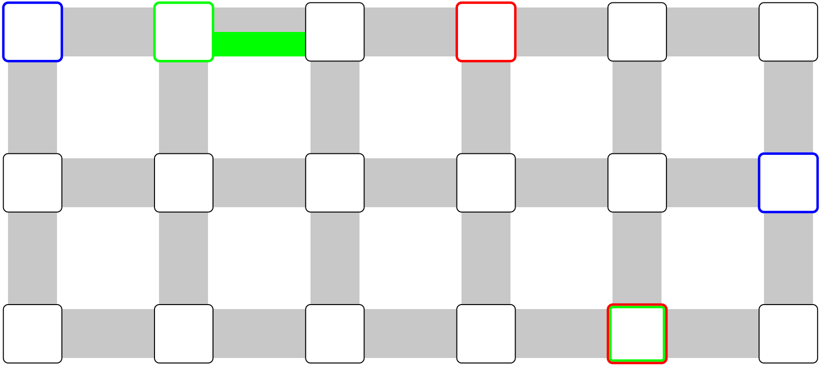
Simple greedy (**SG**)

- Simple greedy (**SG**): greedily assigns communications, hop by hop, on the least loaded link.
- Improved greedy (**IG**): virtually pre-assigns communications onto links, then almost like **SG**.
- Two-bend (**TB**): for each communication, chooses the best path with two bends.
- XY improver (**XYI**): starts from XY assignment, and moves communications from the highest loaded link.
- Path remover (**PR**): virtually pre-assigns communications onto links, and iteratively prevents communications from using highly loaded links.

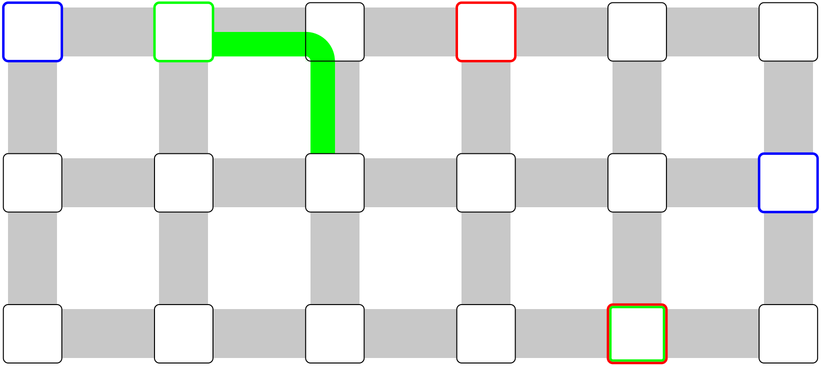
Simple greedy (SG)



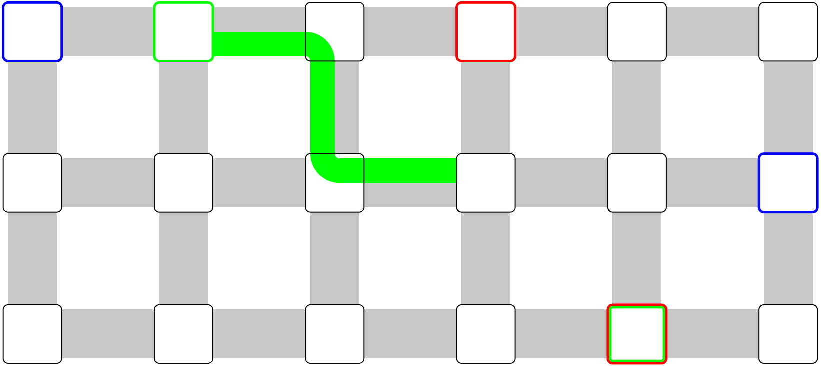
Simple greedy (SG)



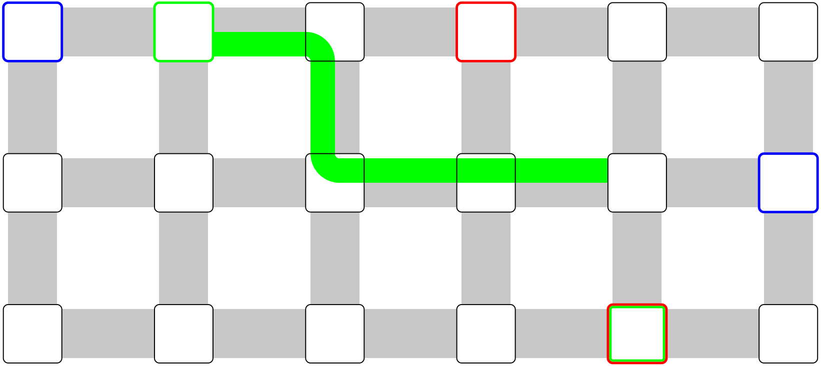
Simple greedy (**SG**)



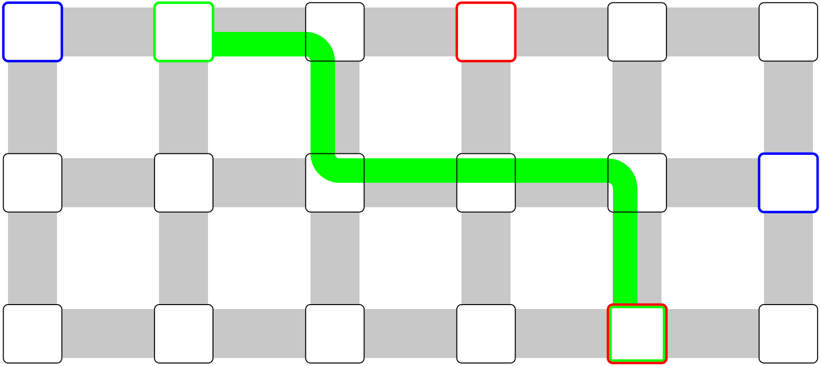
Simple greedy (SG)



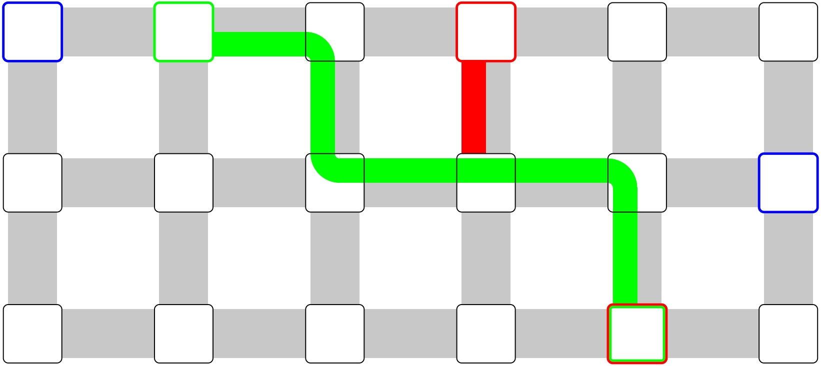
Simple greedy (SG)



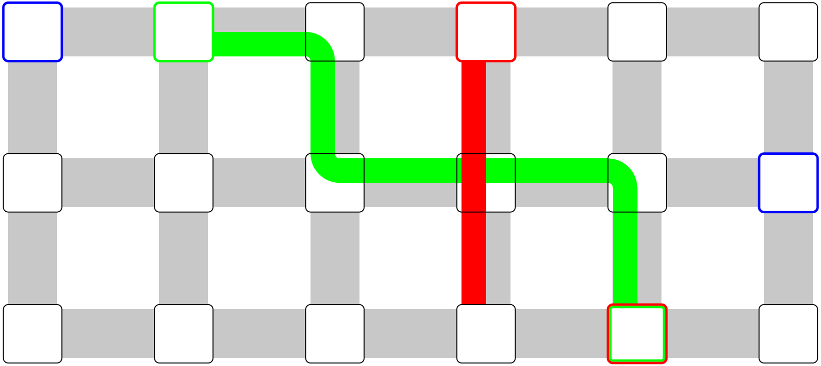
Simple greedy (SG)



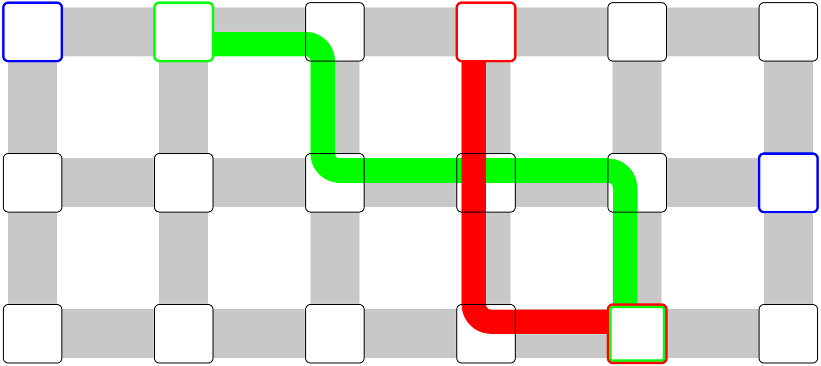
Simple greedy (SG)



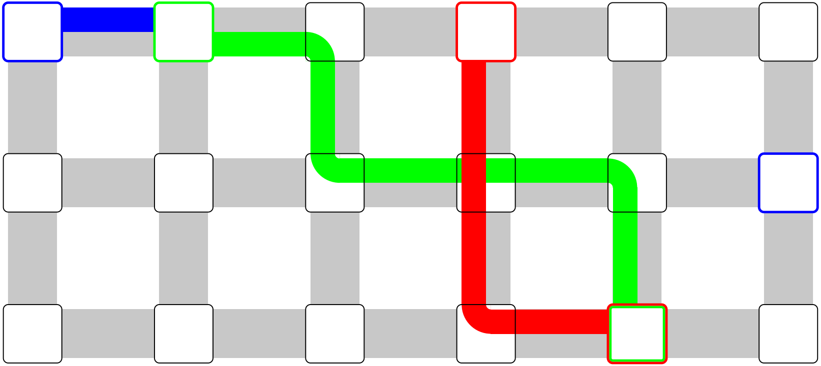
Simple greedy (**SG**)



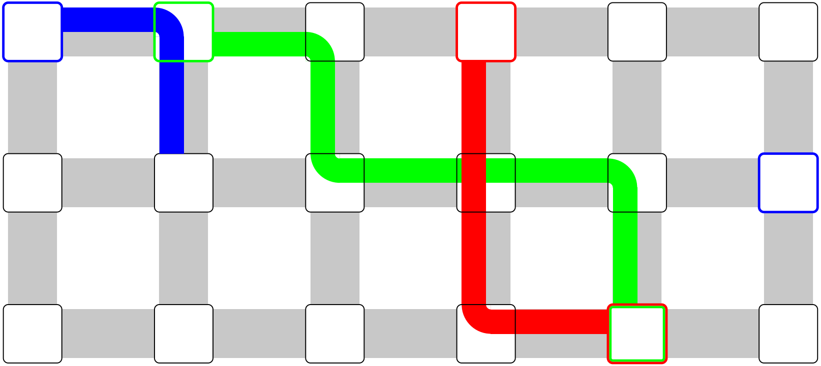
Simple greedy (**SG**)



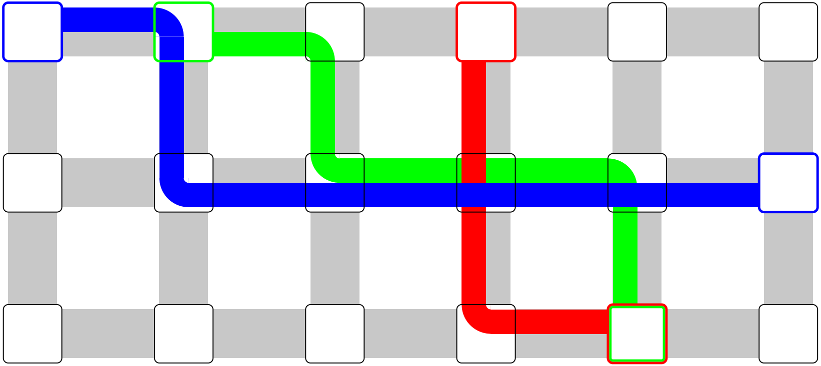
Simple greedy (SG)



Simple greedy (SG)



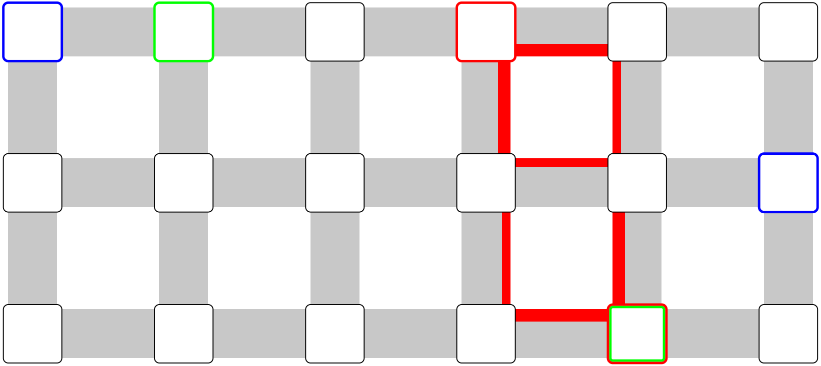
Simple greedy (**SG**)



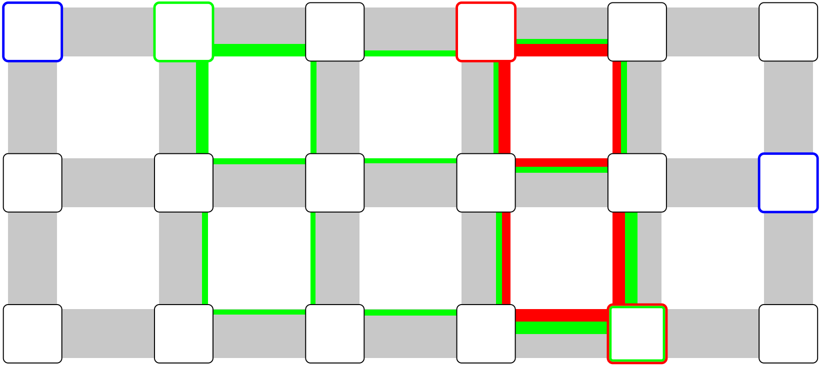
Improved greedy (IG)

- Simple greedy (**SG**): greedily assigns communications, hop by hop, on the least loaded link.
- Improved greedy (**IG**): virtually pre-assigns communications onto links, then almost like **SG**.
- Two-bend (**TB**): for each communication, chooses the best path with two bends.
- XY improver (**XYI**): starts from *XY* assignment, and moves communications from the highest loaded link.
- Path remover (**PR**): virtually pre-assigns communications onto links, and iteratively prevents communications from using highly loaded links.

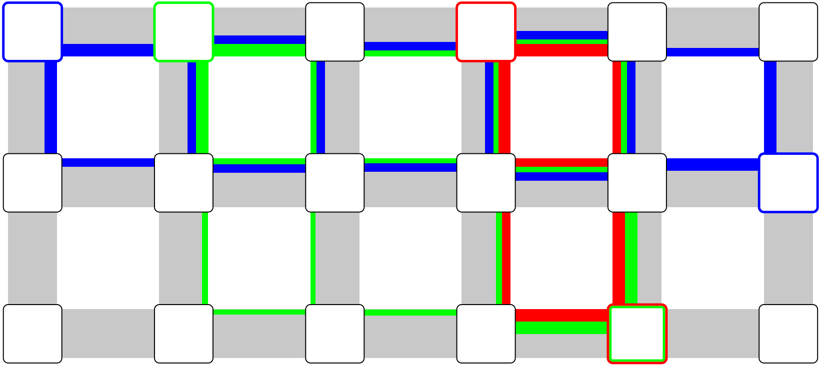
Improved greedy (**IG**)



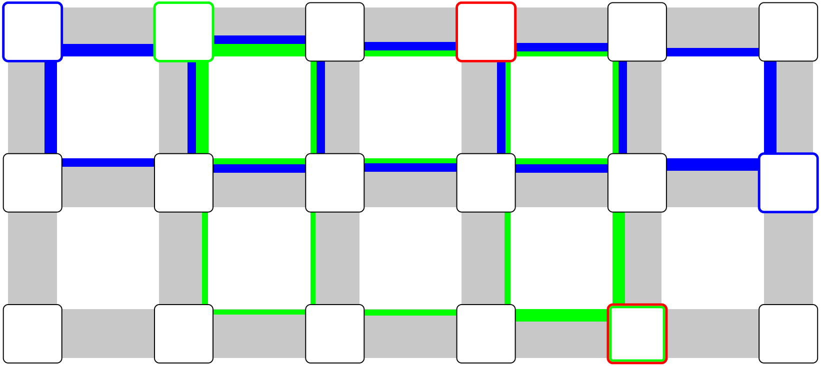
Improved greedy (IG)



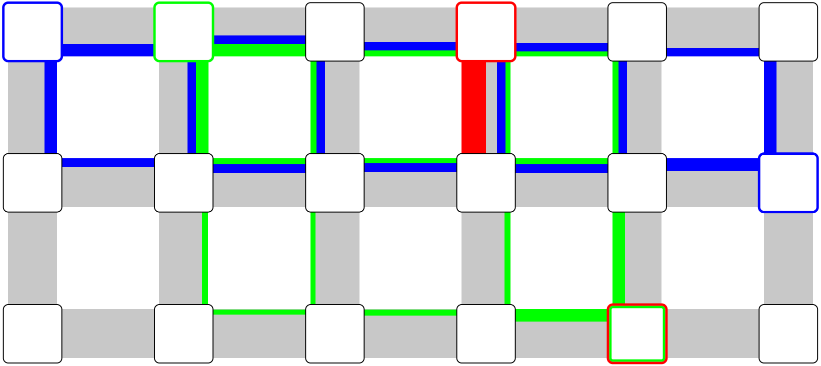
Improved greedy (IG)



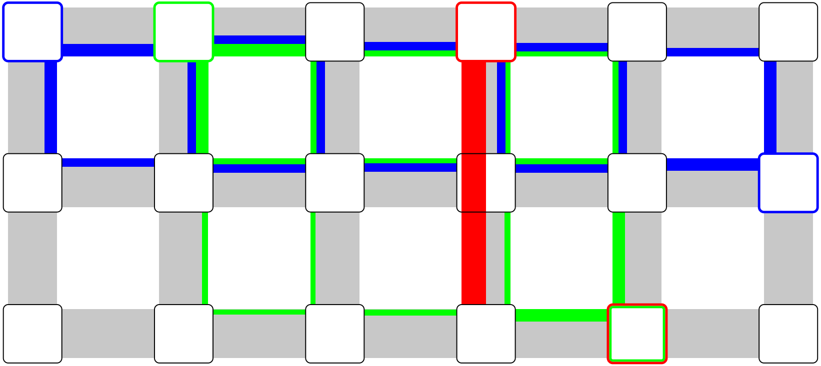
Improved greedy (**IG**)



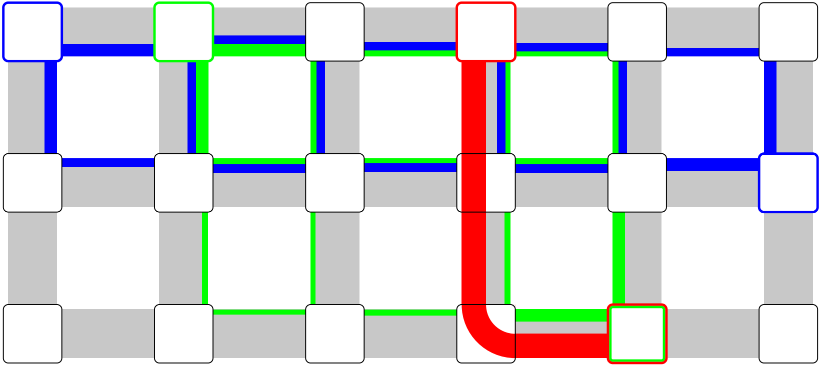
Improved greedy (IG)



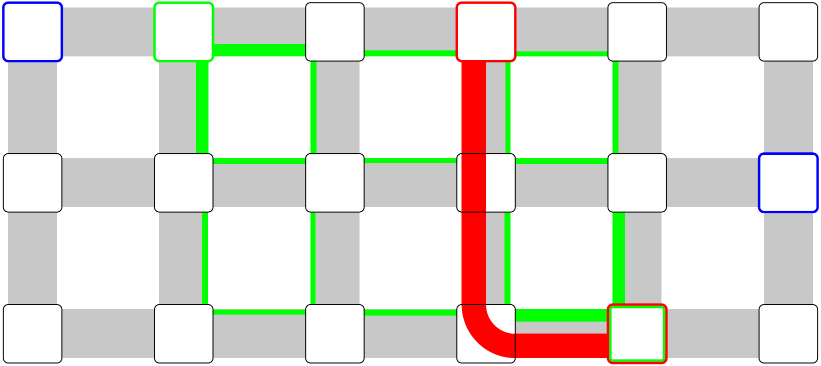
Improved greedy (**IG**)



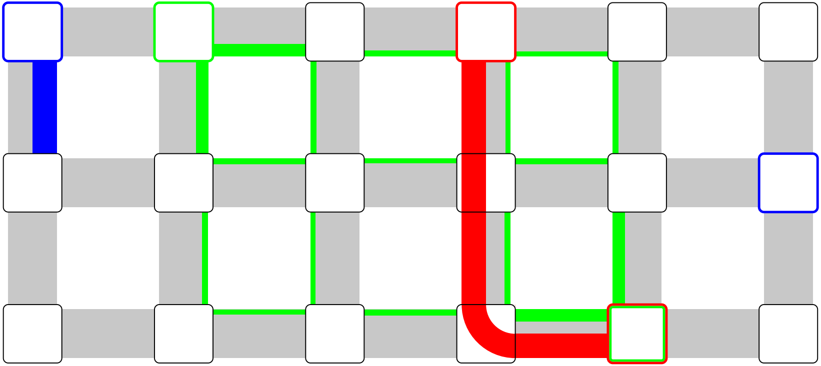
Improved greedy (IG)



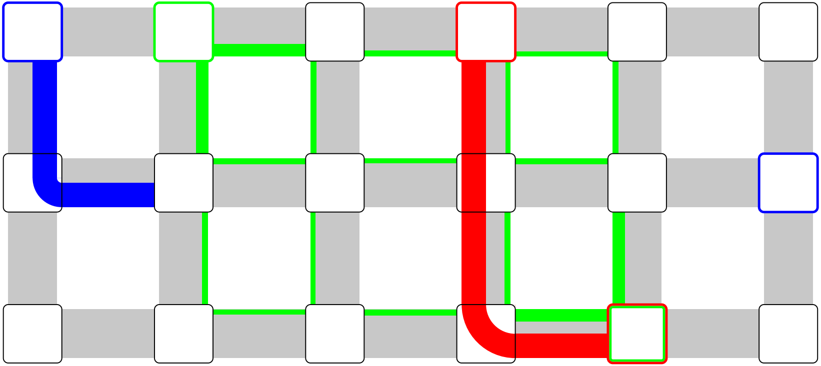
Improved greedy (**IG**)



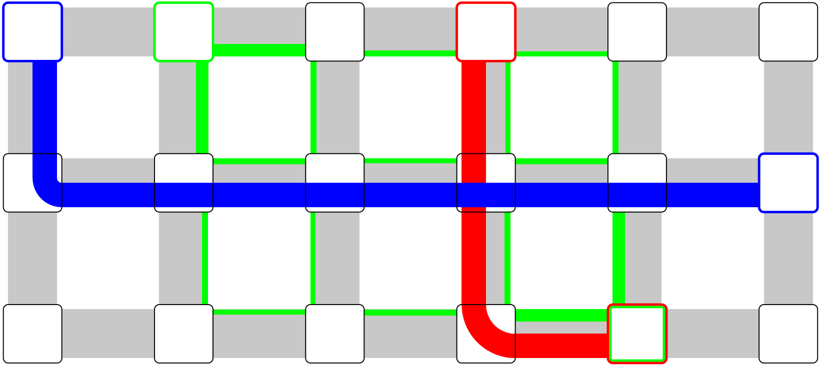
Improved greedy (IG)



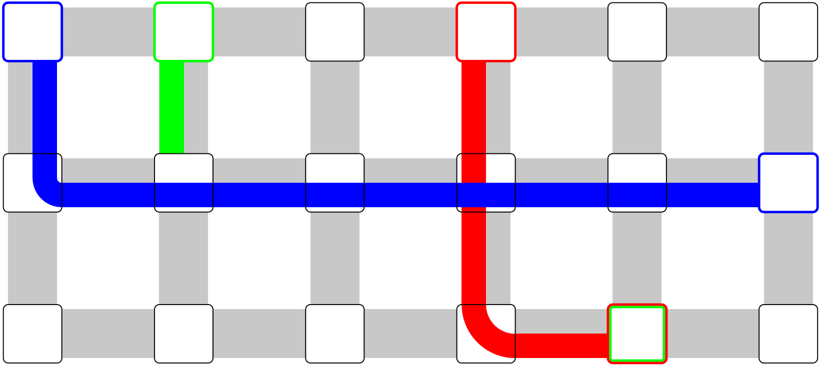
Improved greedy (IG)



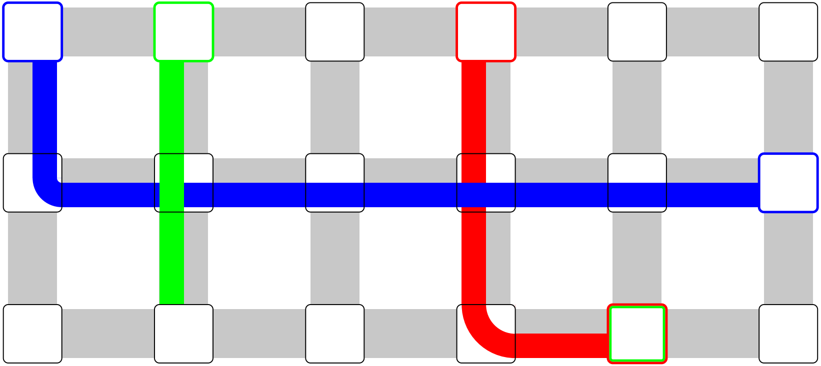
Improved greedy (IG)



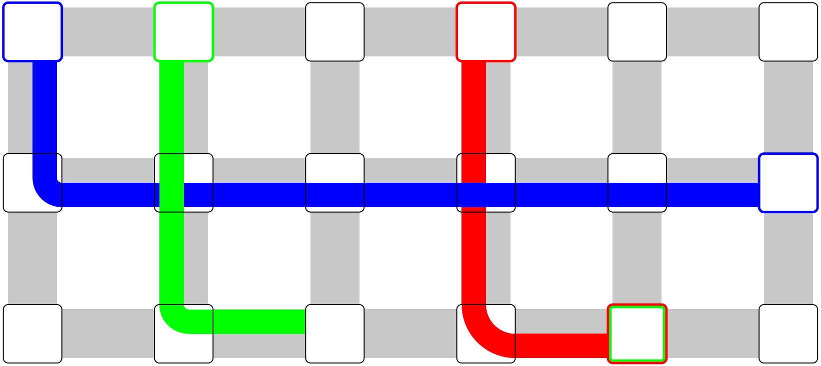
Improved greedy (IG)



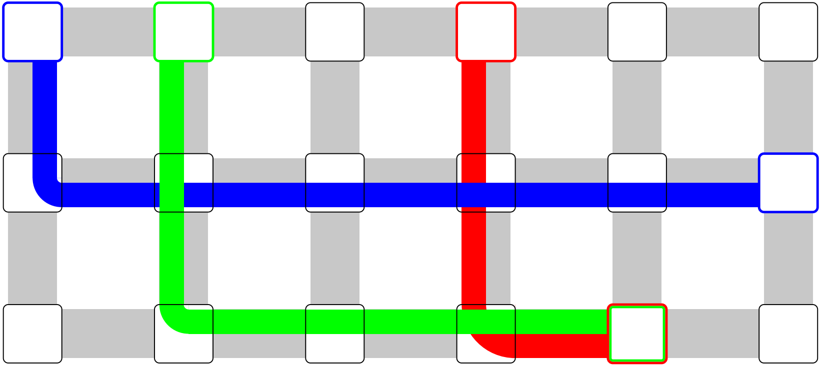
Improved greedy (IG)



Improved greedy (IG)



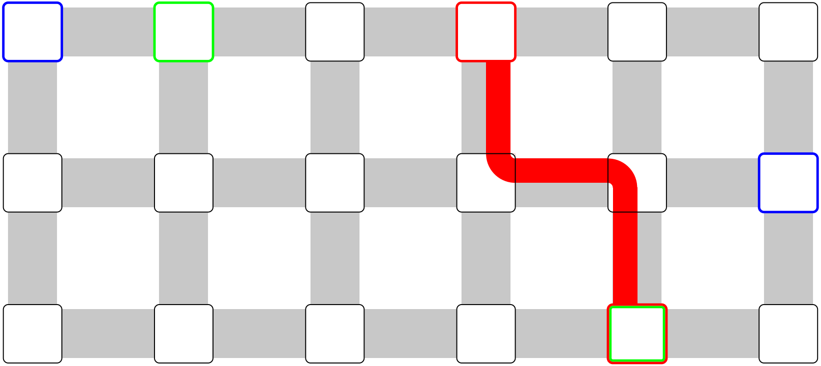
Improved greedy (IG)



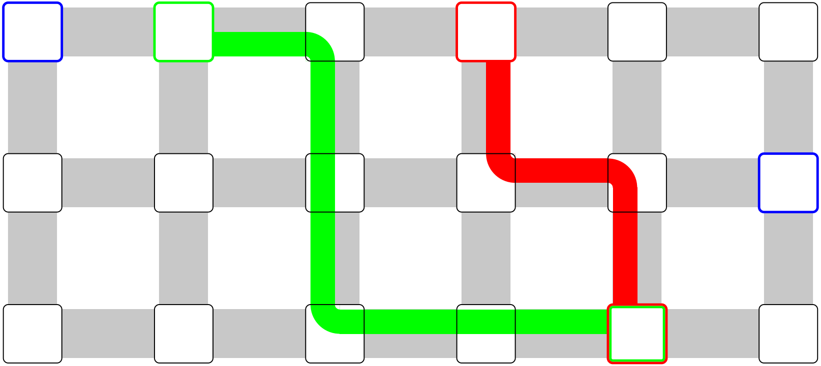
Two-bend (TB)

- Simple greedy (**SG**): greedily assigns communications, hop by hop, on the least loaded link.
- Improved greedy (**IG**): virtually pre-assigns communications onto links, then almost like **SG**.
- Two-bend (**TB**): for each communication, chooses the best path with two bends.
- XY improver (**XYI**): starts from *XY* assignment, and moves communications from the highest loaded link.
- Path remover (**PR**): virtually pre-assigns communications onto links, and iteratively prevents communications from using highly loaded links.

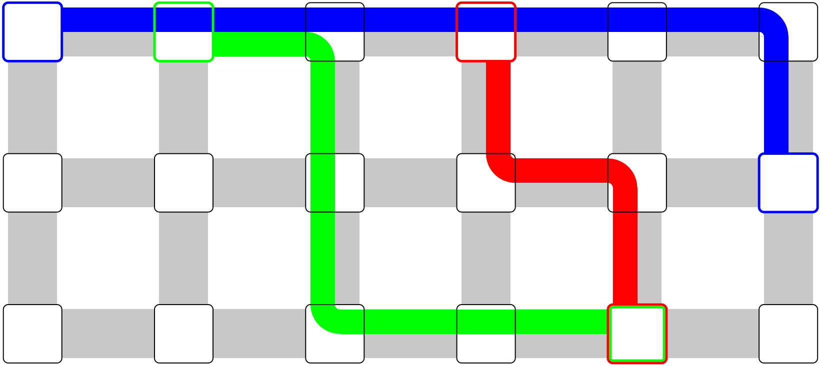
Two-bend (TB)



Two-bend (TB)



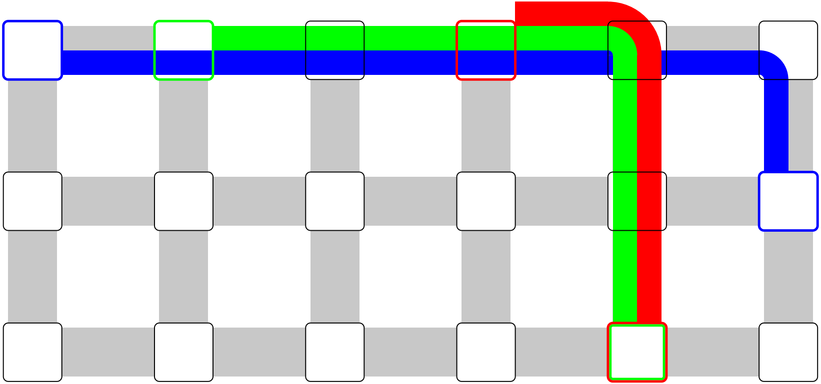
Two-bend (TB)



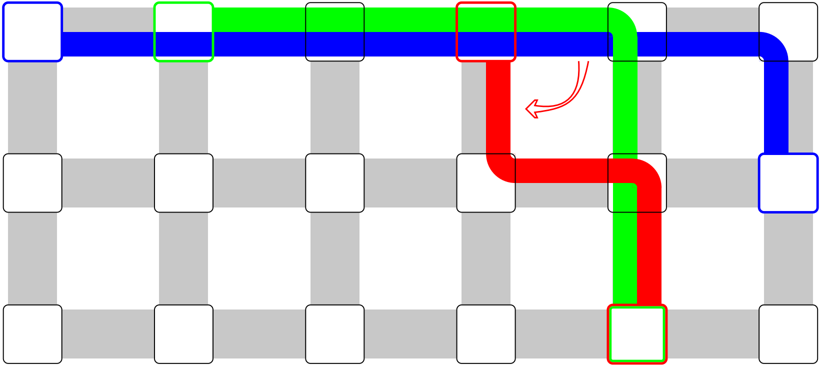
XY improver (XYI)

- Simple greedy (**SG**): greedily assigns communications, hop by hop, on the least loaded link.
- Improved greedy (**IG**): virtually pre-assigns communications onto links, then almost like **SG**.
- Two-bend (**TB**): for each communication, chooses the best path with two bends.
- XY improver (**XYI**): starts from XY assignment, and moves communications from the highest loaded link.
- Path remover (**PR**): virtually pre-assigns communications onto links, and iteratively prevents communications from using highly loaded links.

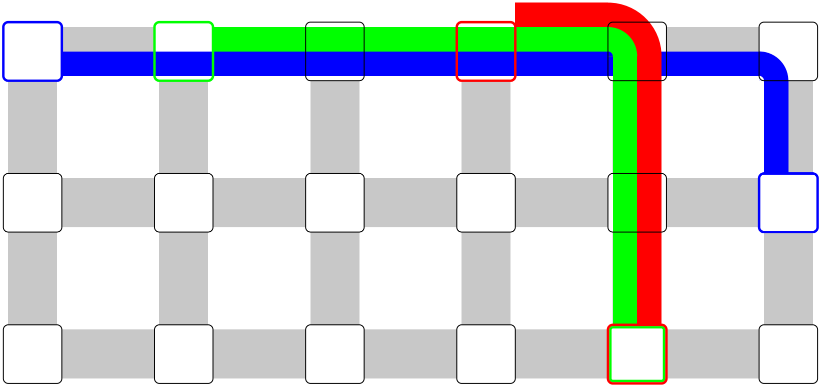
XY improver (XYI)



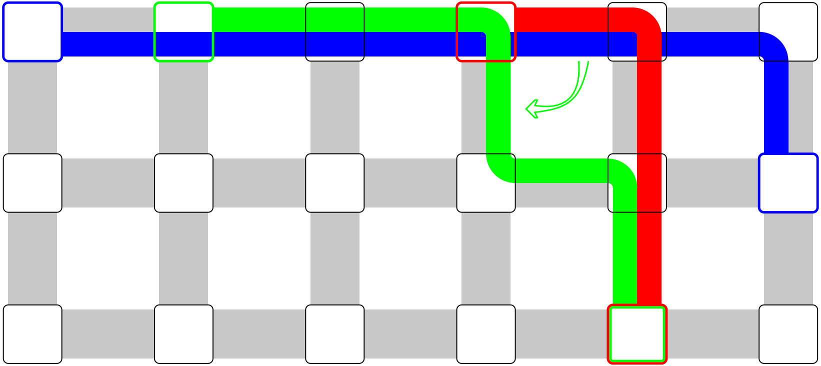
XY improver (**XYI**)



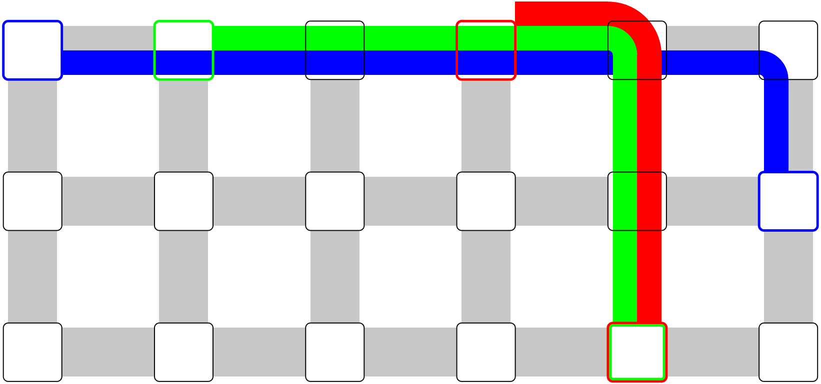
XY improver (**XYI**)



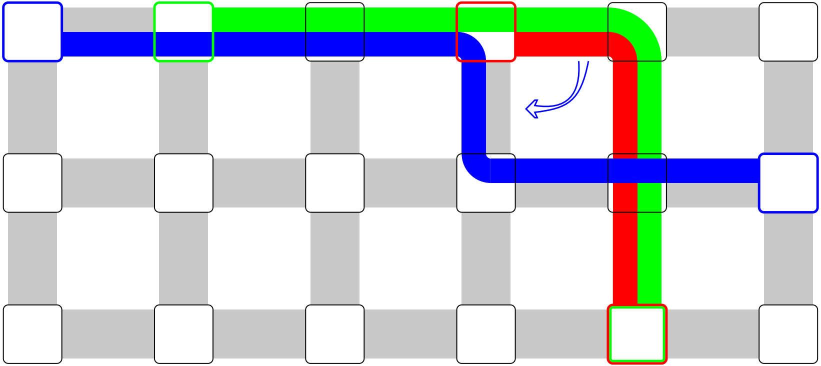
XY improver (XYI)



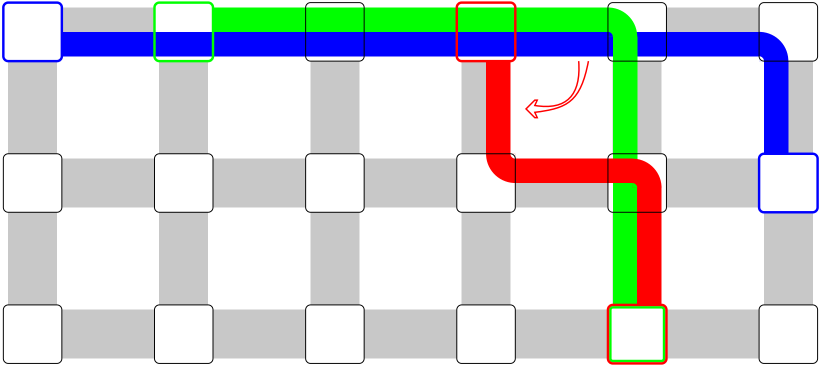
XY improver (**XYI**)



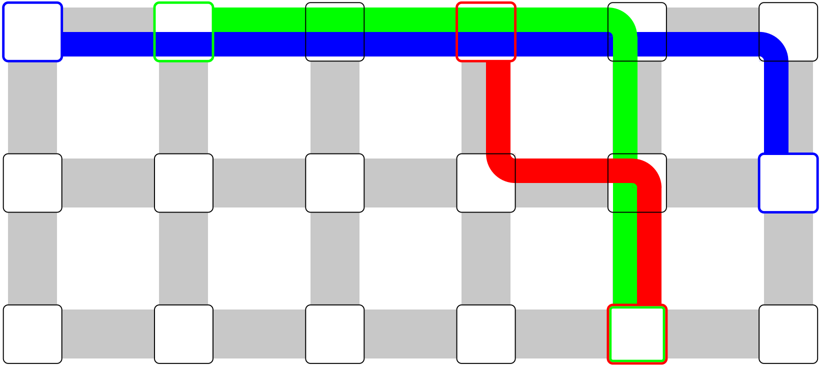
XY improver (XYI)



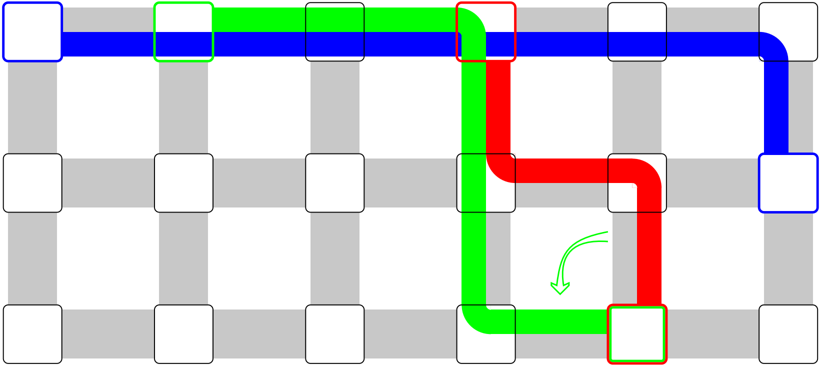
XY improver (**XYI**)



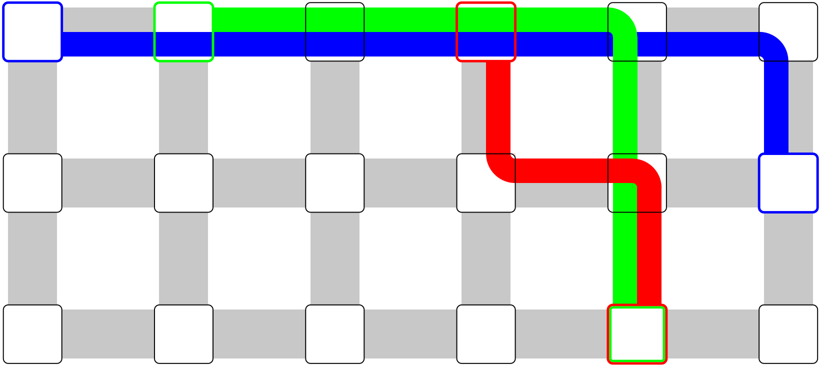
XY improver (XYI)



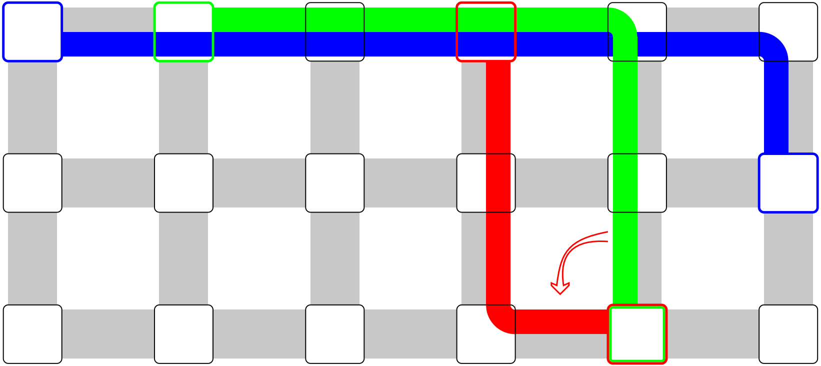
XY improver (**XYI**)



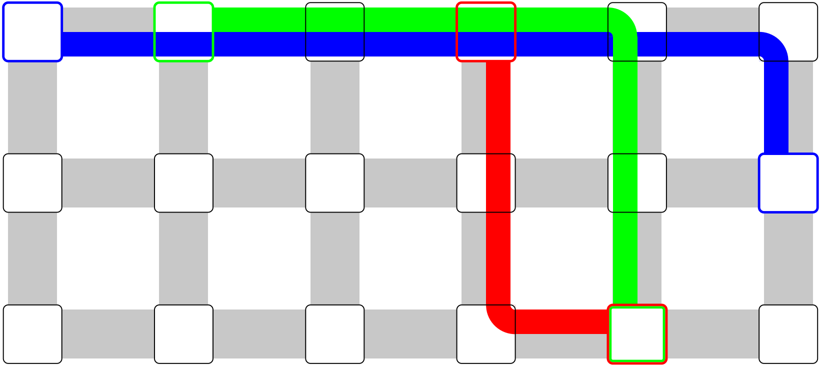
XY improver (**XYI**)



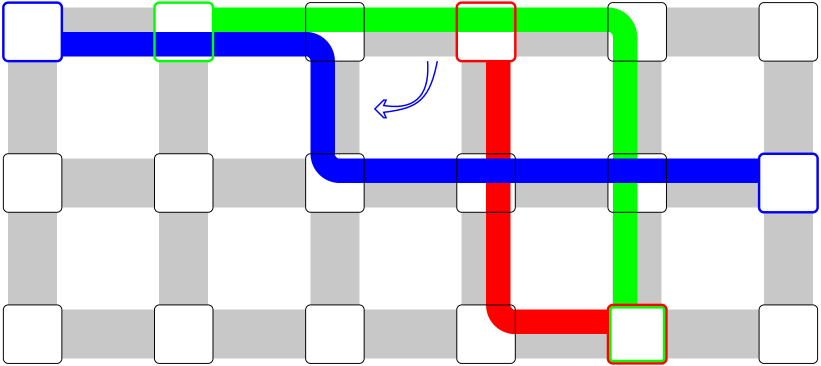
XY improver (**XYI**)



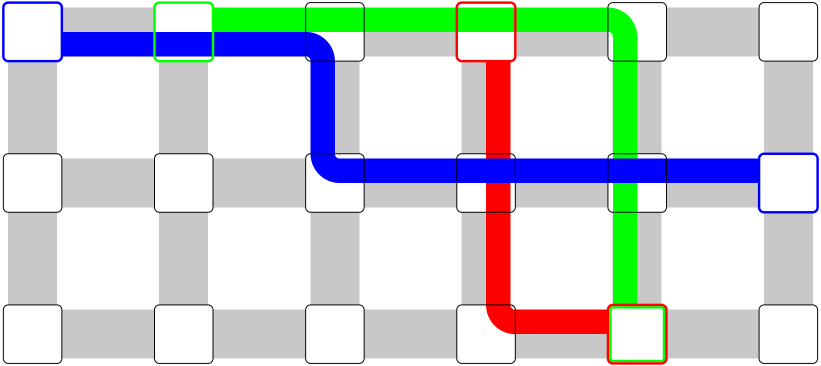
XY improver (XYI)



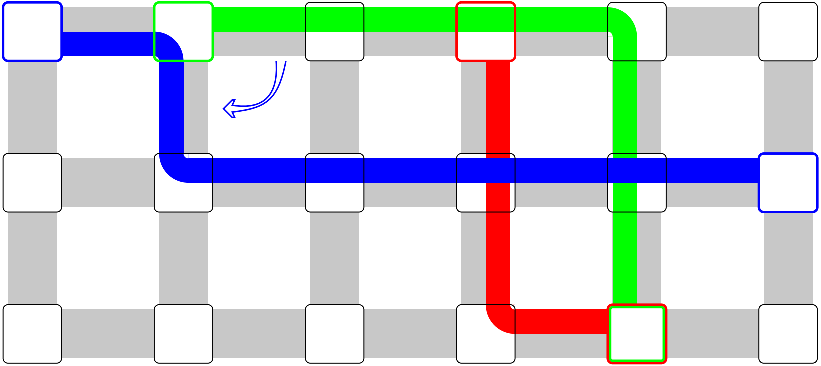
XY improver (**XYI**)



XY improver (**XYI**)



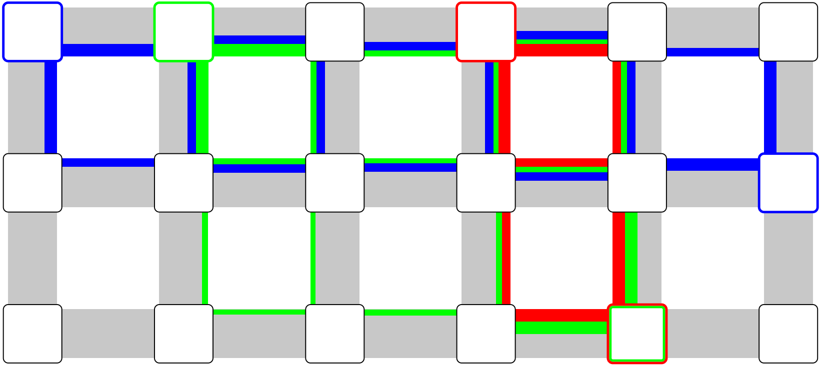
XY improver (XYI)



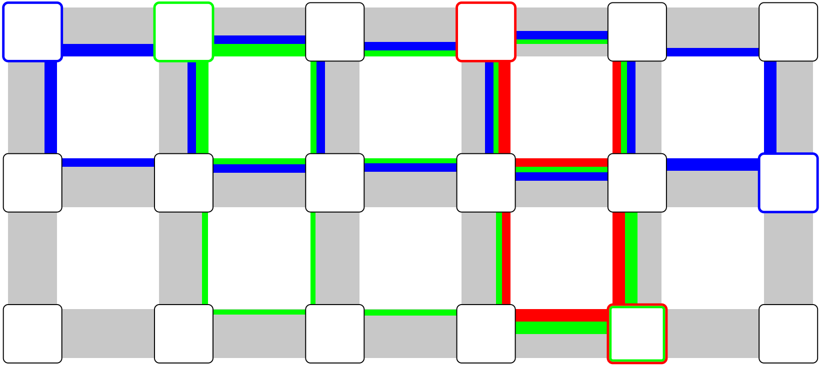
Path remover (**PR**)

- Simple greedy (**SG**): greedily assigns communications, hop by hop, on the least loaded link.
- Improved greedy (**IG**): virtually pre-assigns communications onto links, then almost like **SG**.
- Two-bend (**TB**): for each communication, chooses the best path with two bends.
- XY improver (**XYI**): starts from *XY* assignment, and moves communications from the highest loaded link.
- Path remover (**PR**): virtually pre-assigns communications onto links, and iteratively prevents communications from using highly loaded links.

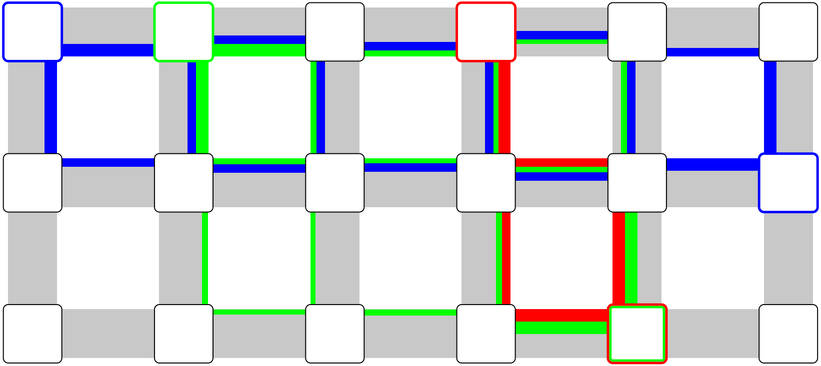
Path remover (PR)



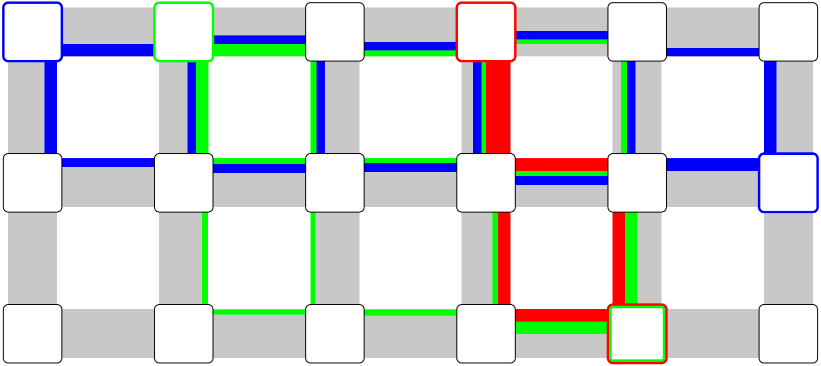
Path remover (PR)



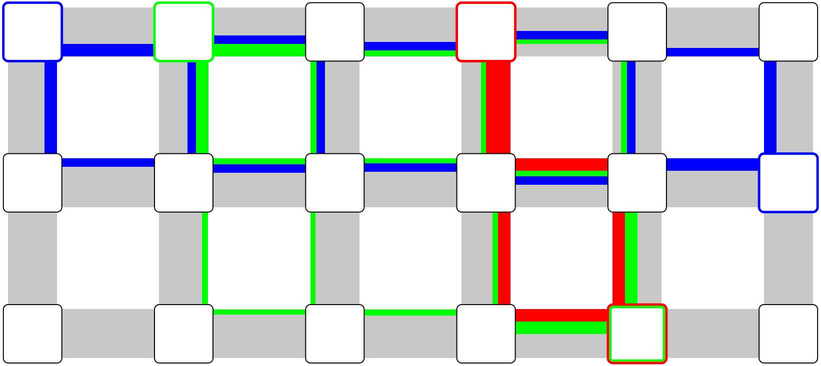
Path remover (PR)



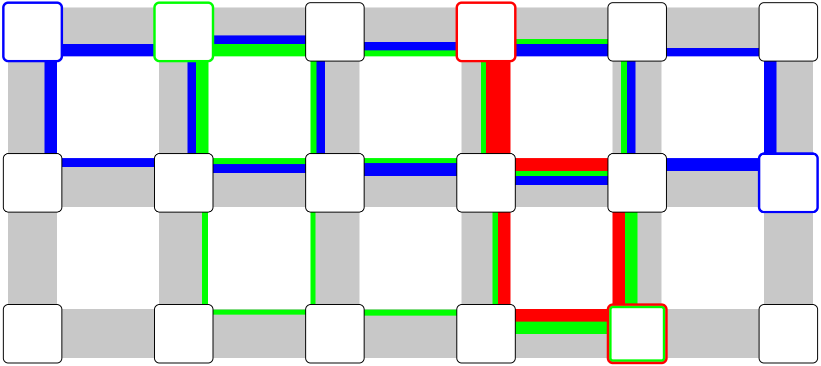
Path remover (PR)



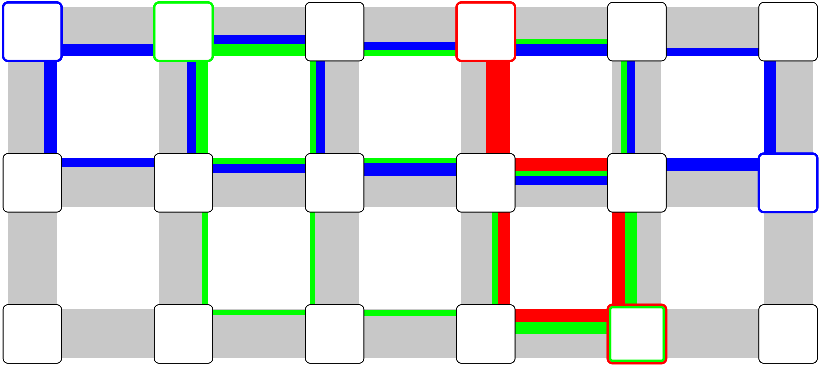
Path remover (PR)



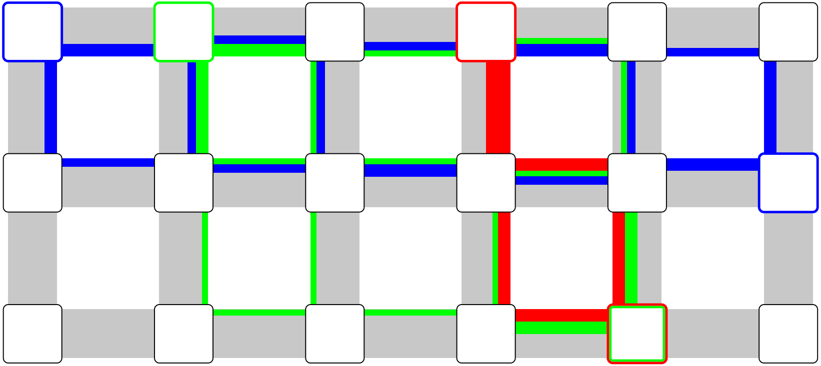
Path remover (PR)



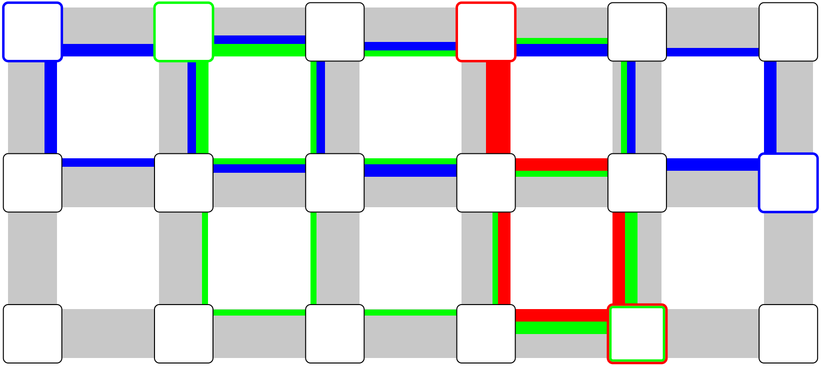
Path remover (PR)



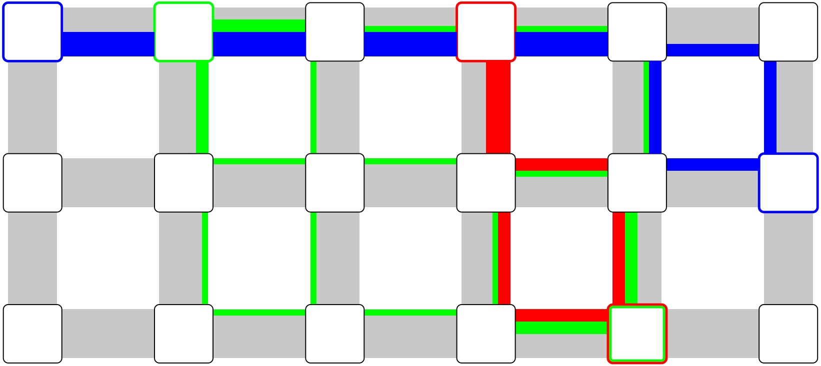
Path remover (PR)



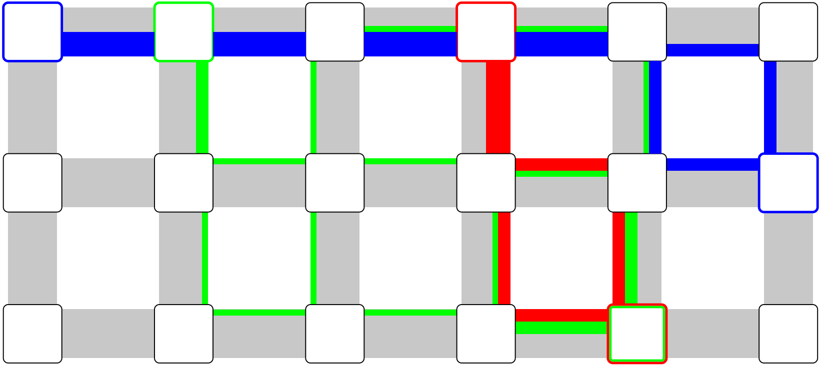
Path remover (**PR**)



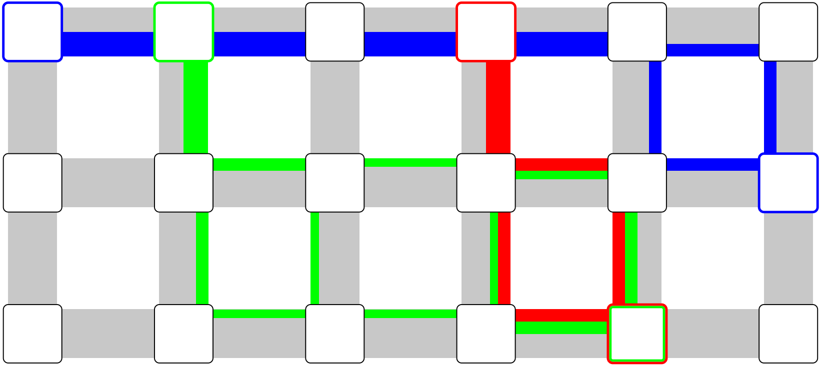
Path remover (PR)



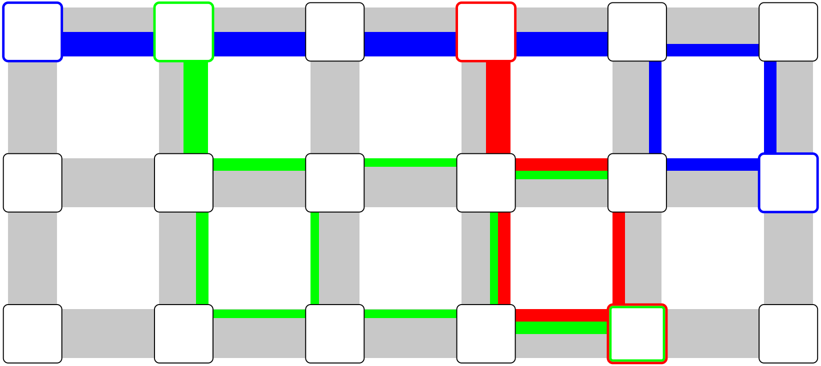
Path remover (PR)



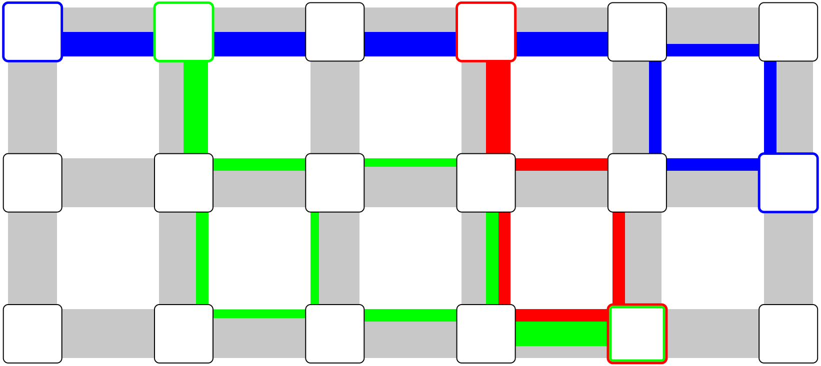
Path remover (PR)



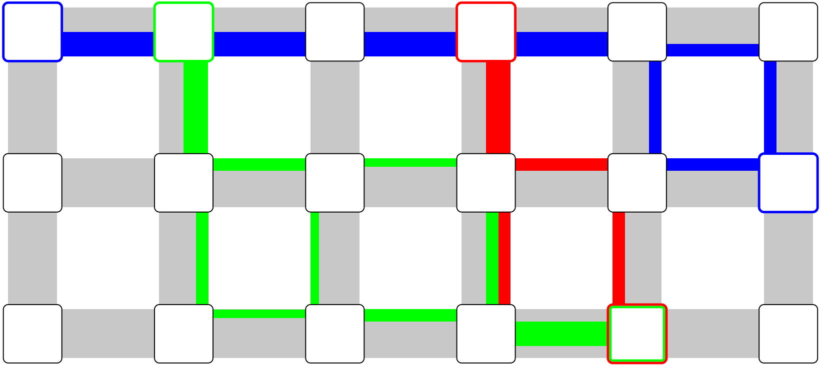
Path remover (**PR**)



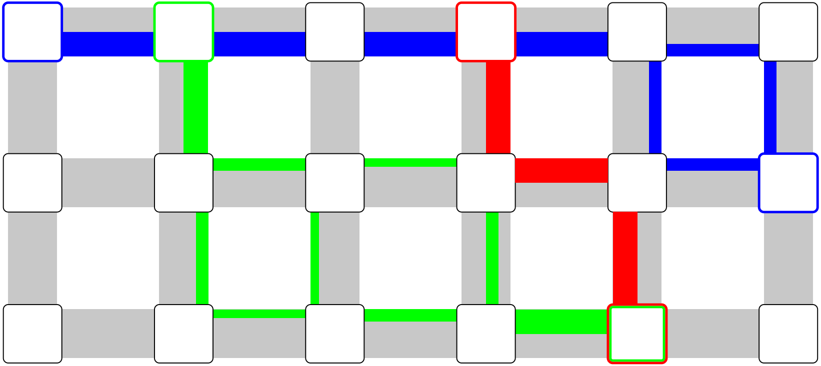
Path remover (PR)



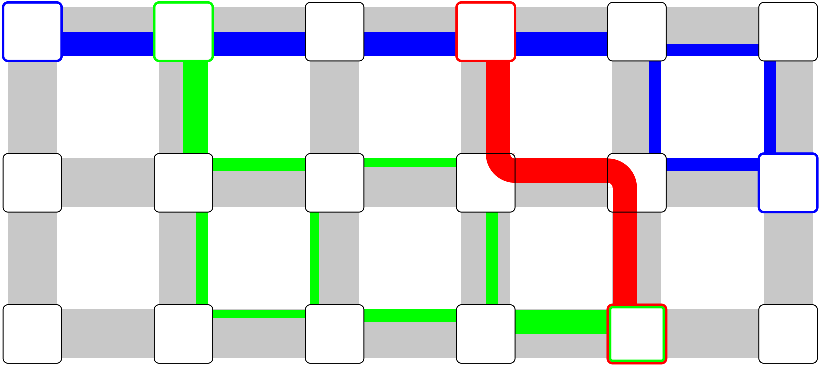
Path remover (PR)



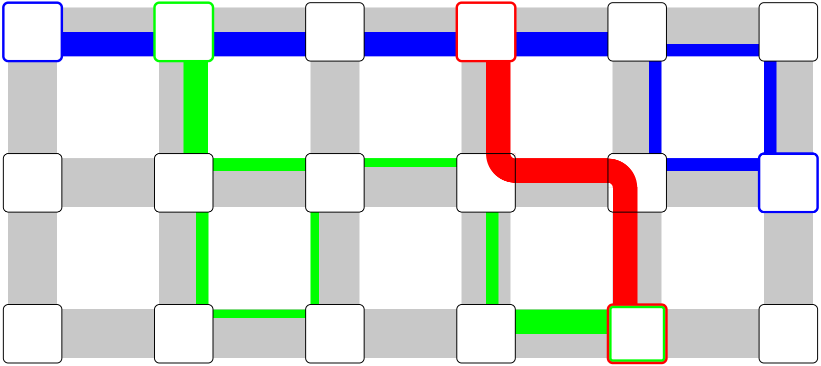
Path remover (PR)



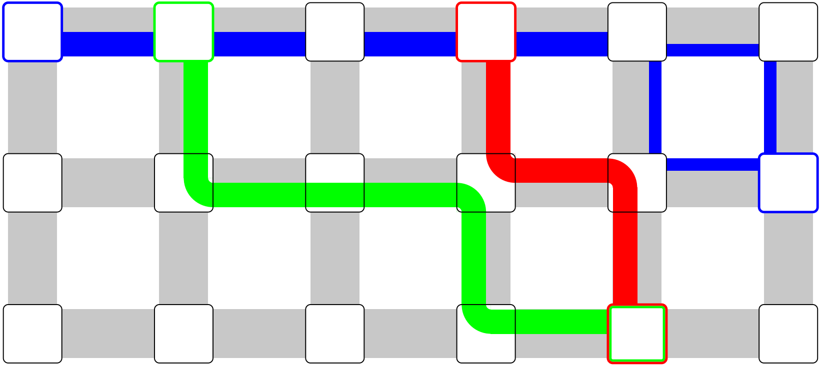
Path remover (PR)



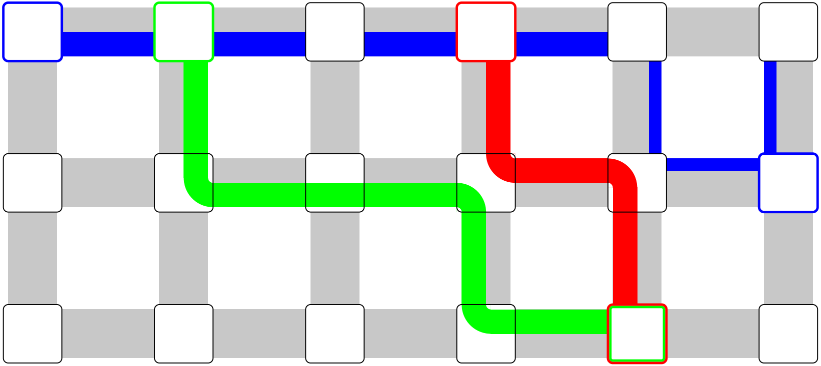
Path remover (PR)



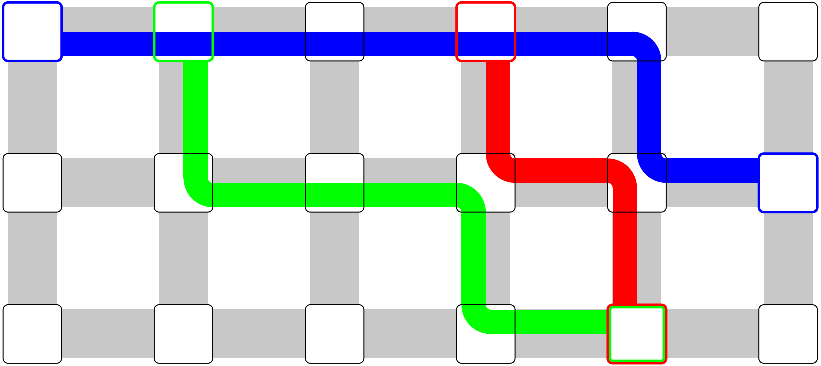
Path remover (PR)



Path remover (PR)



Path remover (PR)



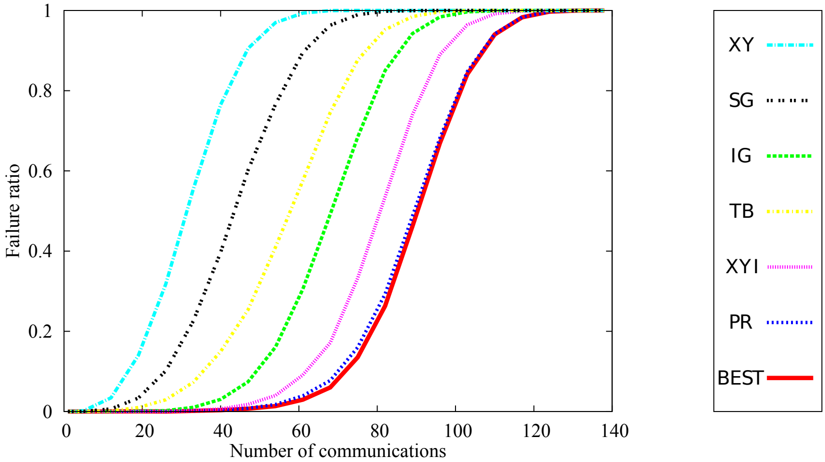
Outline of the talk

Simulation settings

- 8×8 CMP
- Discrete frequencies: $1\ \mathrm{Gb/s}$, $2.5\ \mathrm{Gb/s}$ and $3.5\ \mathrm{Gb/s}$
- $P_{\mathrm{leak}} = 16.9\,\mathrm{mW}$, $P_0 = 5.41$ and $\alpha = 2.95$
- Random source and sink nodes for the communications

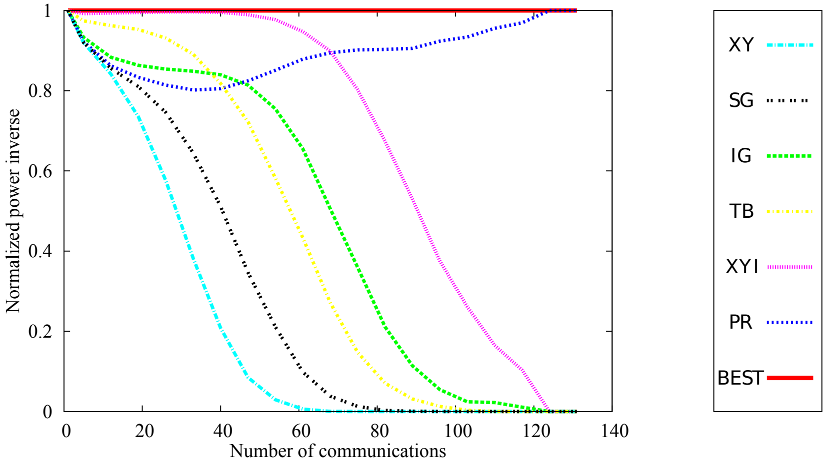
- **BEST** heuristic: best heuristic among all five ones on the given problem instance
- Each point of the graph: average on 50000 sets of communications

Sensitivity to the number of communications



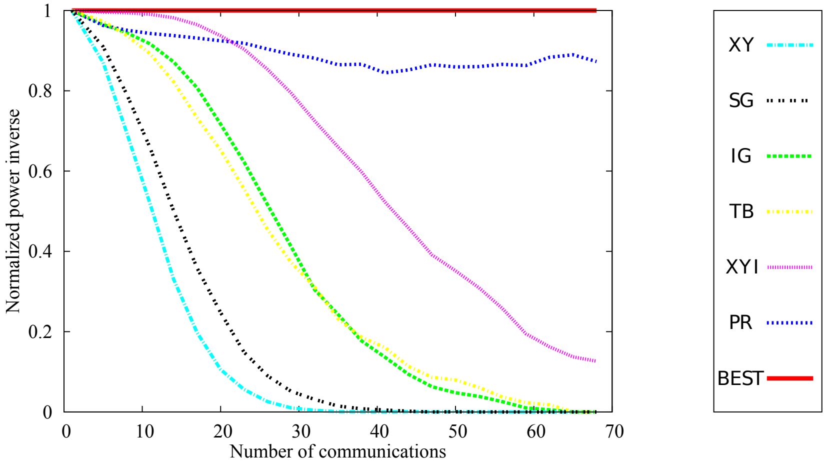
$$100\,\mathrm{Mb/s} \leq \delta_i \leq 1500\,\mathrm{Mb/s}$$

Sensitivity to the number of communications



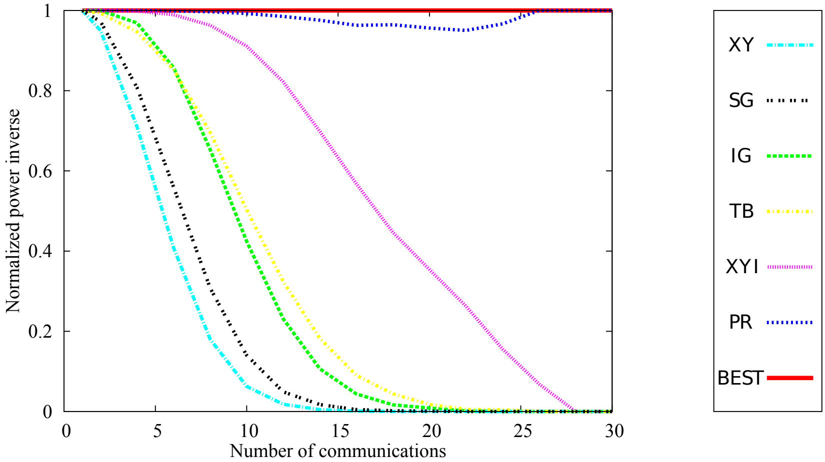
$$100\,\mathrm{Mb/s} \leq \delta_i \leq 1500\,\mathrm{Mb/s}$$

Sensitivity to the number of communications



$100\,\mathrm{Mb/s} \leq \delta_i \leq 2500\,\mathrm{Mb/s}$

Sensitivity to the number of communications



$2500\,\mathrm{Mb/s} \leq \delta_i \leq 3500\,\mathrm{Mb/s}$

Conclusion

- NP-completeness of the problem
- Minimum upper bound of the ratio of the power consumed by an XY routing over the power consumed by a Manhattan routing
- Several single-path heuristics: more solutions and less power consumption
- Future work:
  - Worst case for single-path Manhattan routing when single source and destination
  - Approximation algorithms
  - Optimal solution for single-path Manhattan routings
  - Multi-path heuristics