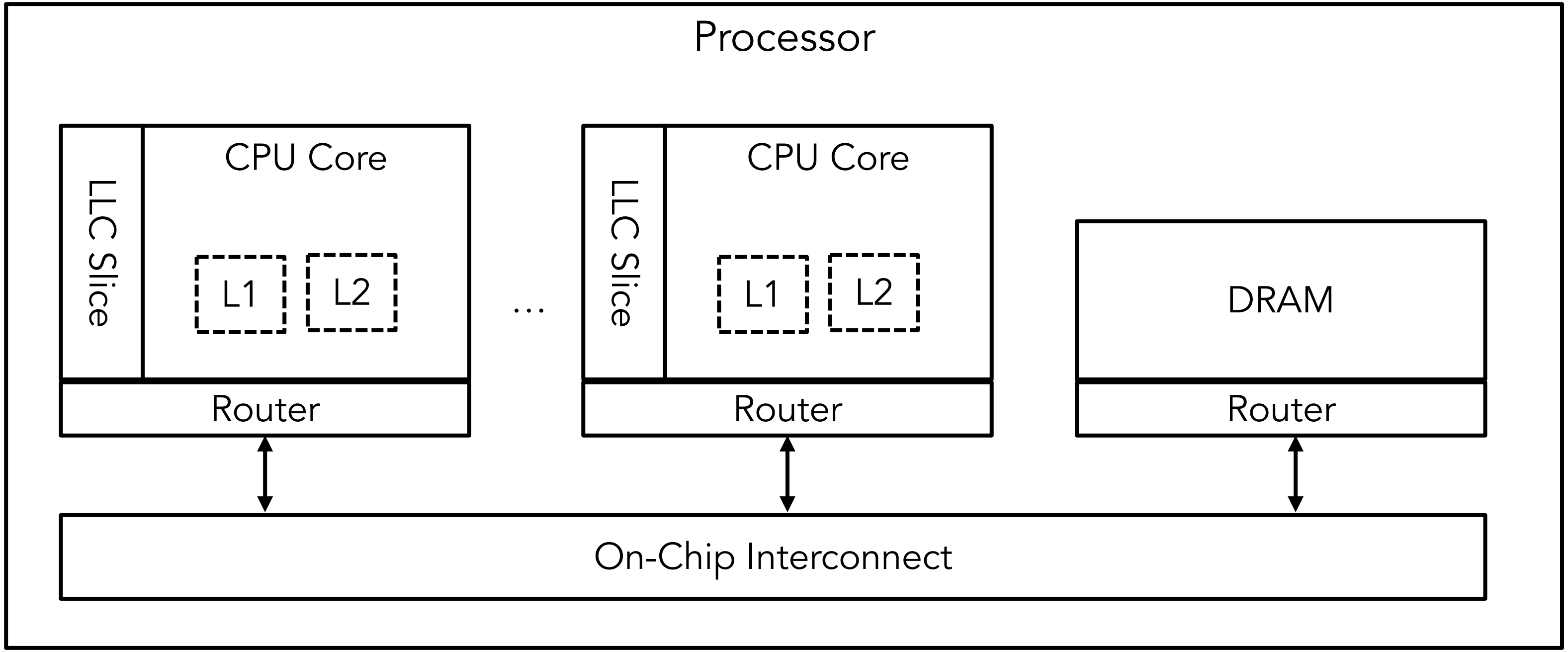
Don't Mesh Around: Side-Channel Attacks and Mitigations on Mesh Interconnects

Miles Dai*, Riccardo Paccagnella*, Miguel Gomez-Garcia,
John McCalpin, Mengjia Yan

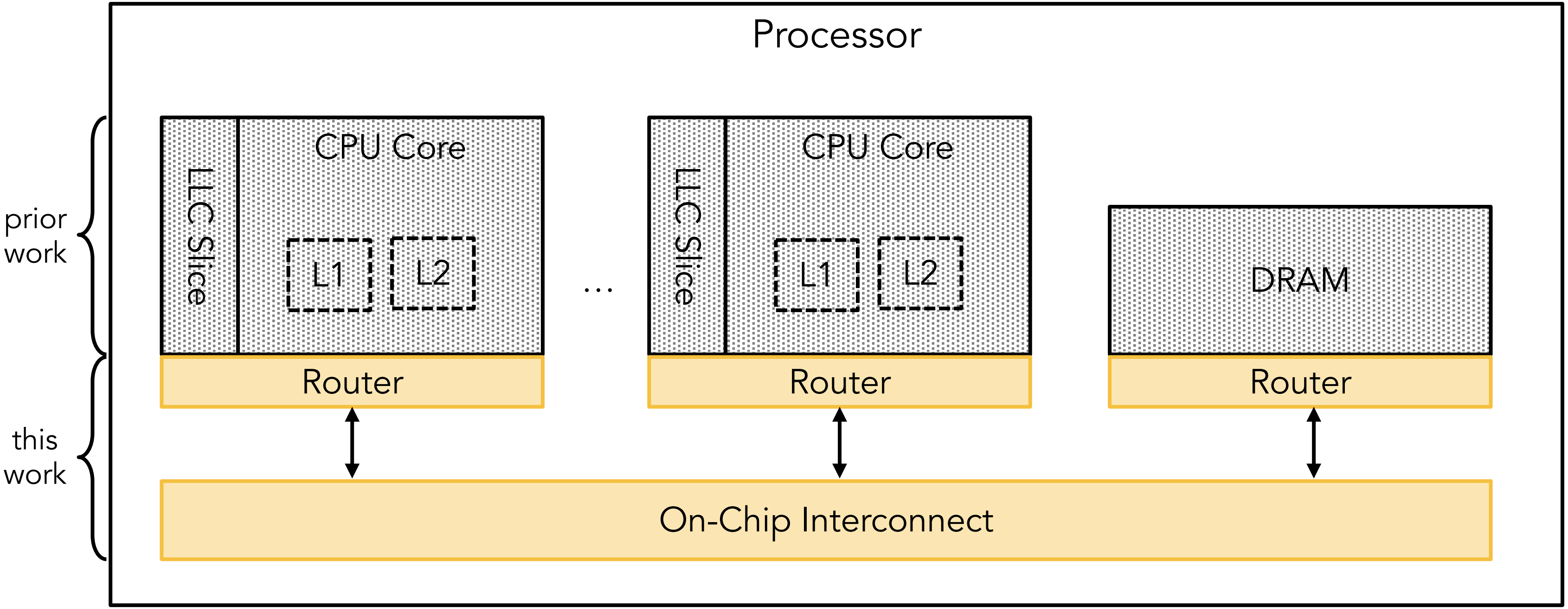
MIT ILLINOIS

(*co-first authors)

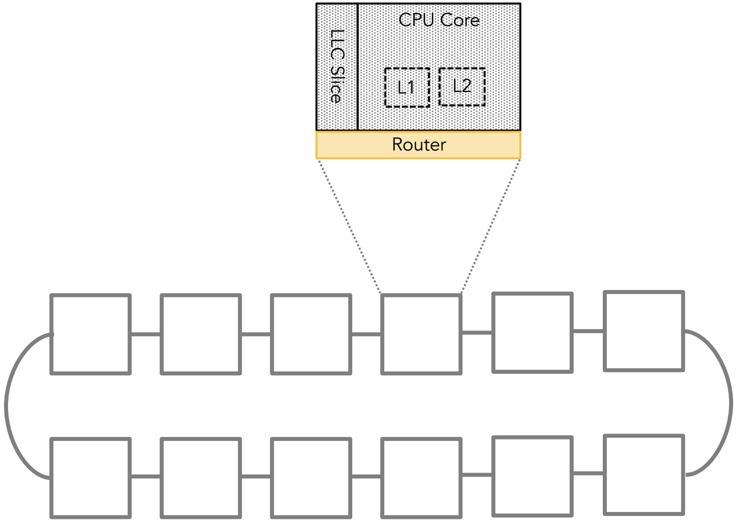
Microarchitectural Attack Surfaces



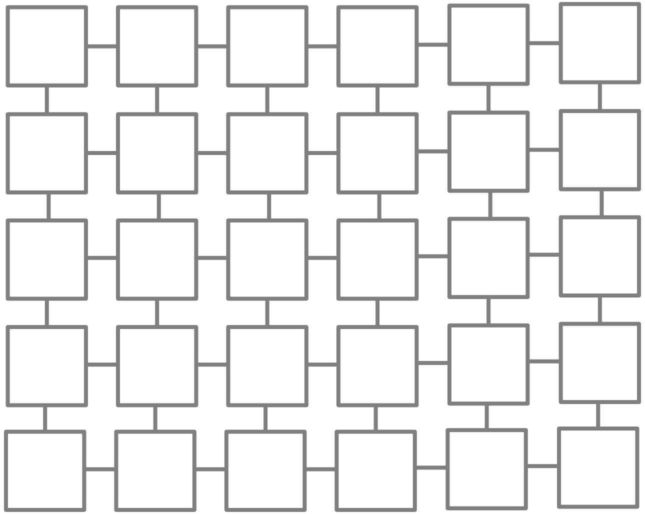
Microarchitectural Attack Surfaces



On-Chip Interconnects

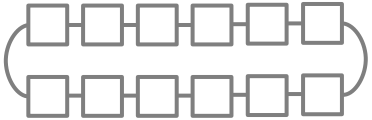


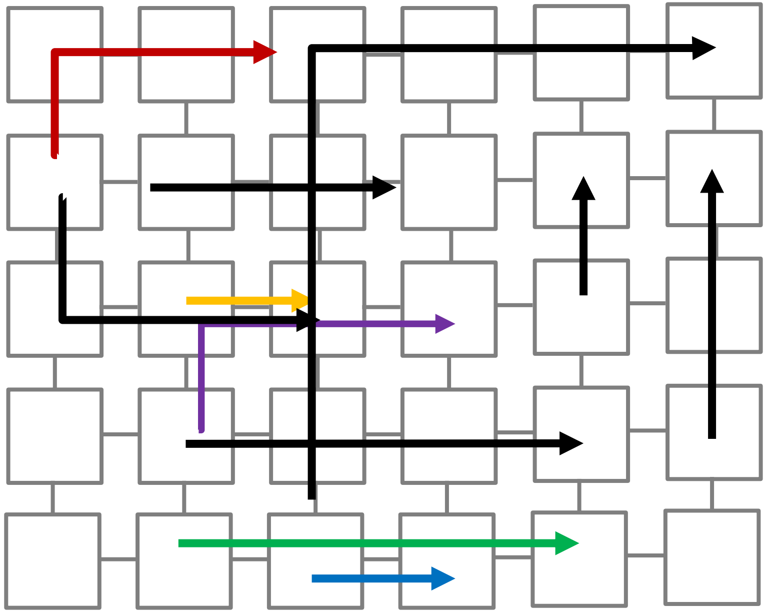
Ring Interconnect



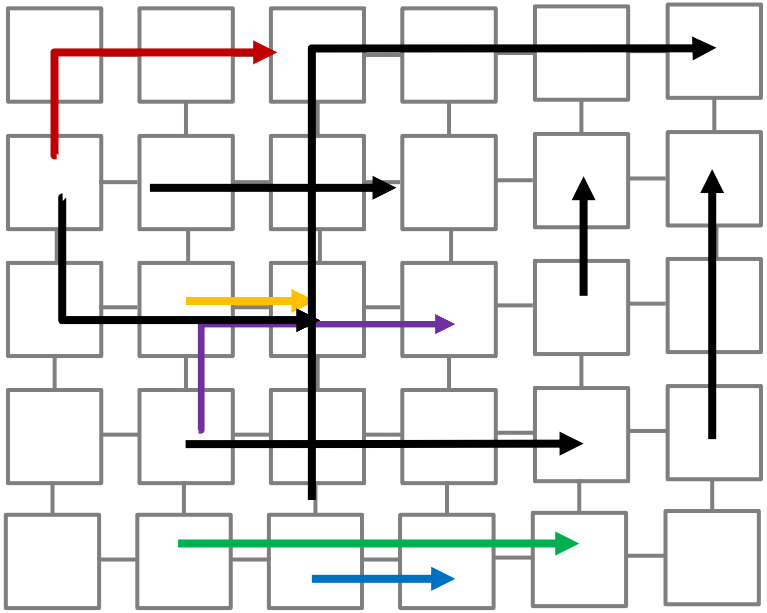
Mesh Interconnect

Mesh Interconnect Challenges





Research Questions



- Is it feasible to construct attacks by only exploiting contention on a mesh interconnect?

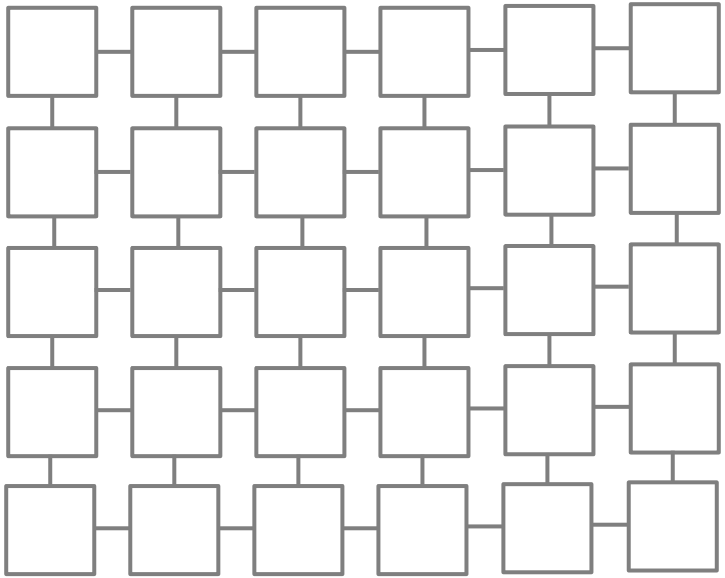
- Are there non-invasive approaches that can mitigate these attacks without requiring hardware modifications?

Reverse Engineering Methodology

Goal: understand how traffic flows interfere with one another

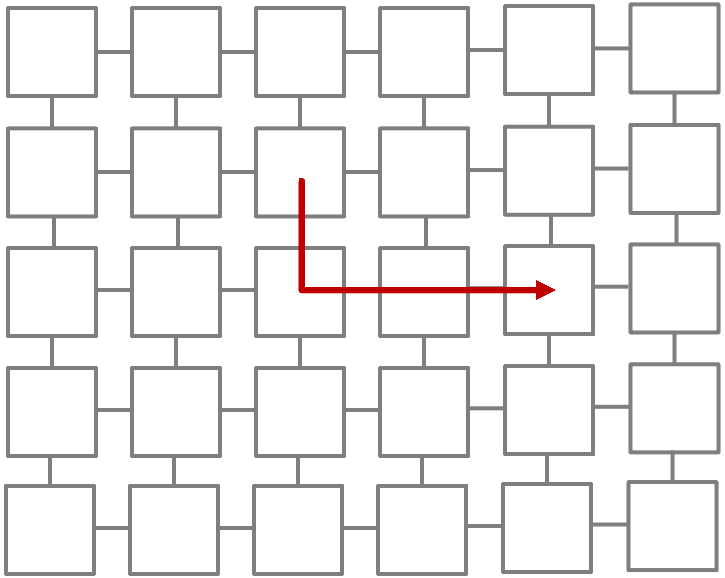
Reverse Engineering Methodology

Goal: understand how traffic flows interfere with one another



Reverse Engineering Methodology

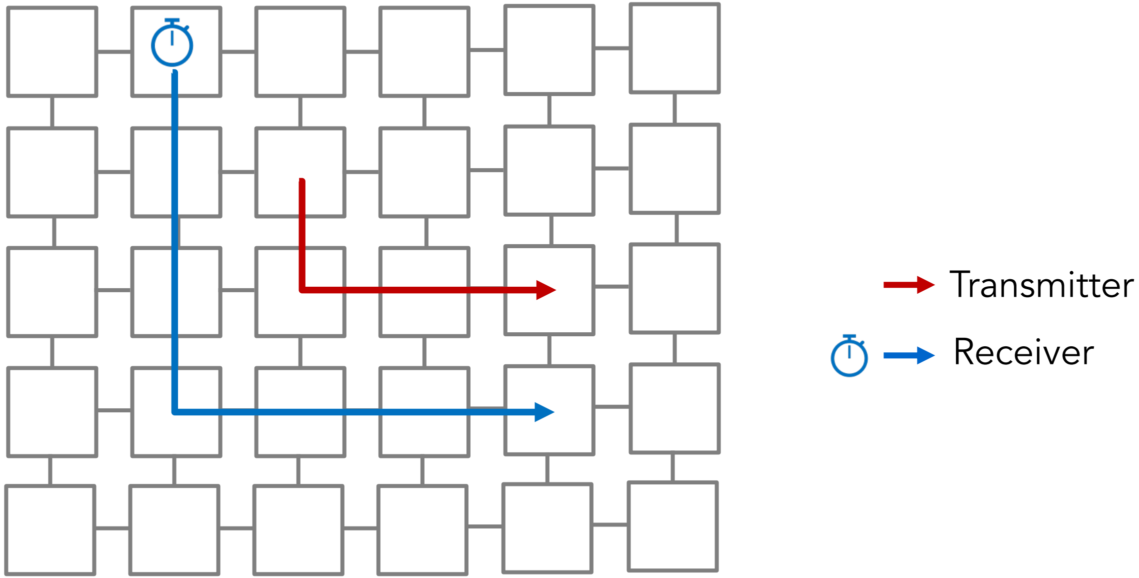
Goal: understand how traffic flows interfere with one another



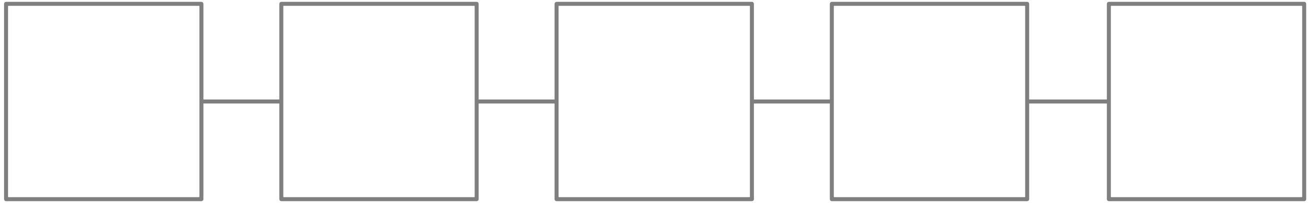
→ Transmitter

Reverse Engineering Methodology

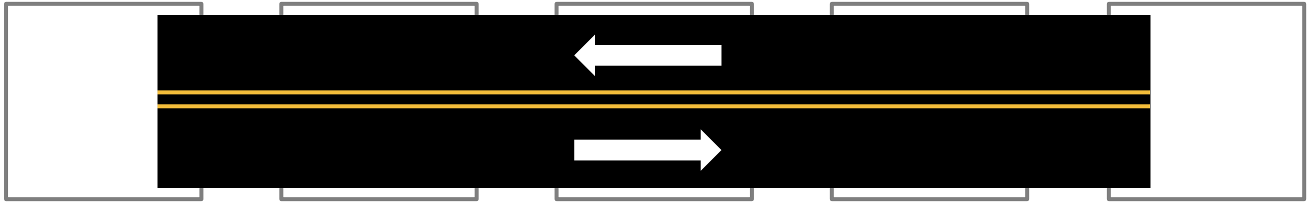
Goal: understand how traffic flows interfere with one another



Conditions for Contention



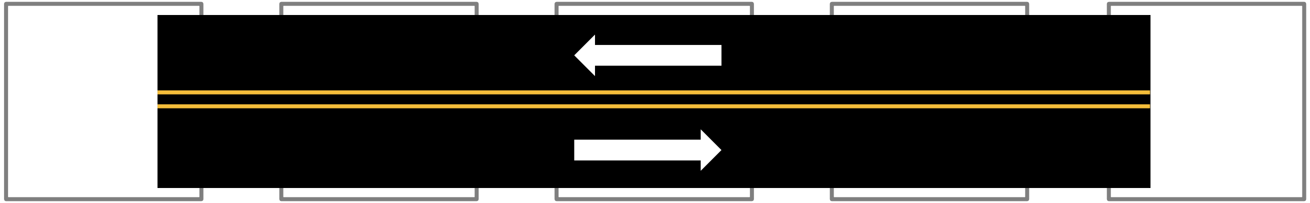
Conditions for Contention



Conditions for Contention

Overlapping paths

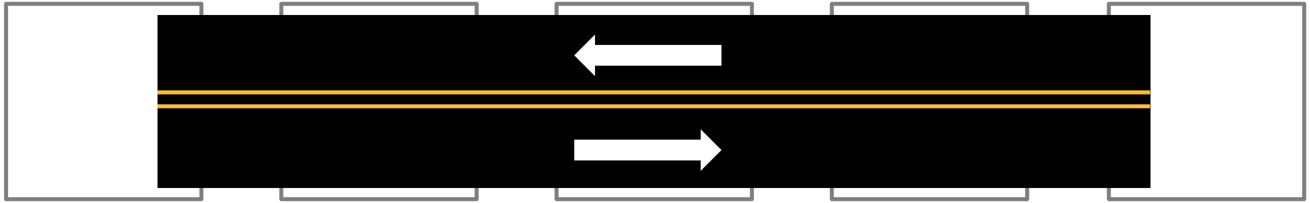
Same direction



Conditions for Contention

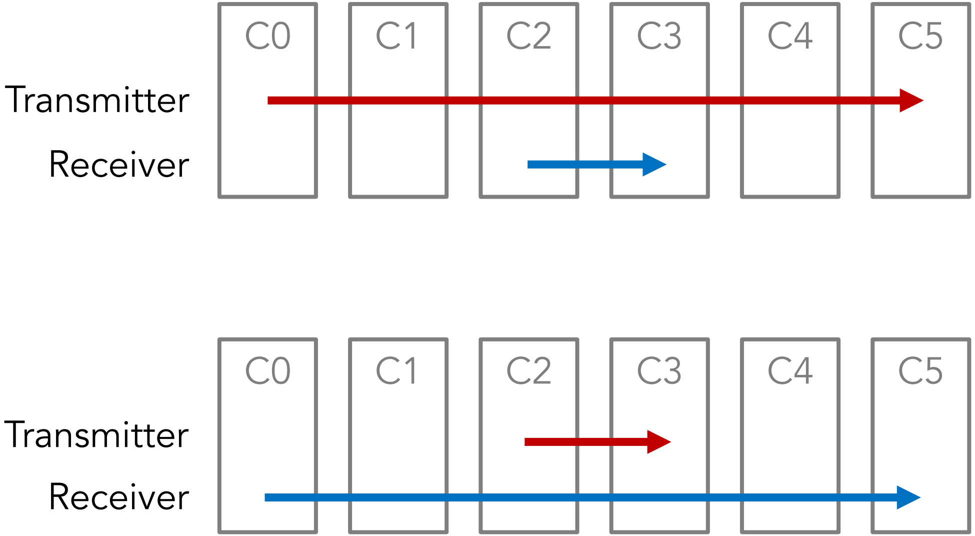
Overlapping paths

Same direction

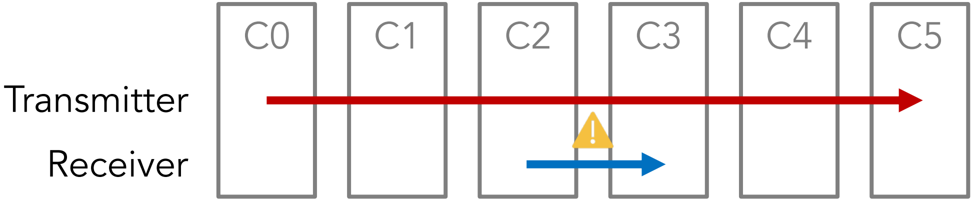


In practice, overlapping flows in same direction do not always cause contention!

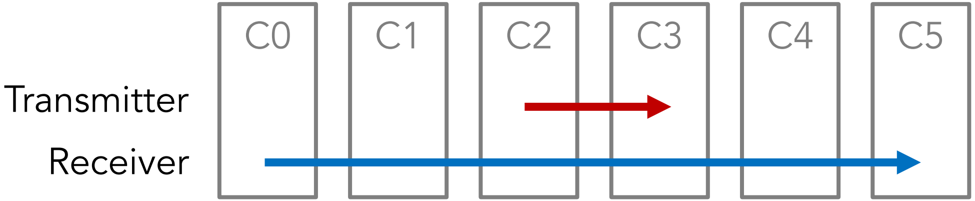
Priority Arbitration



Priority Arbitration

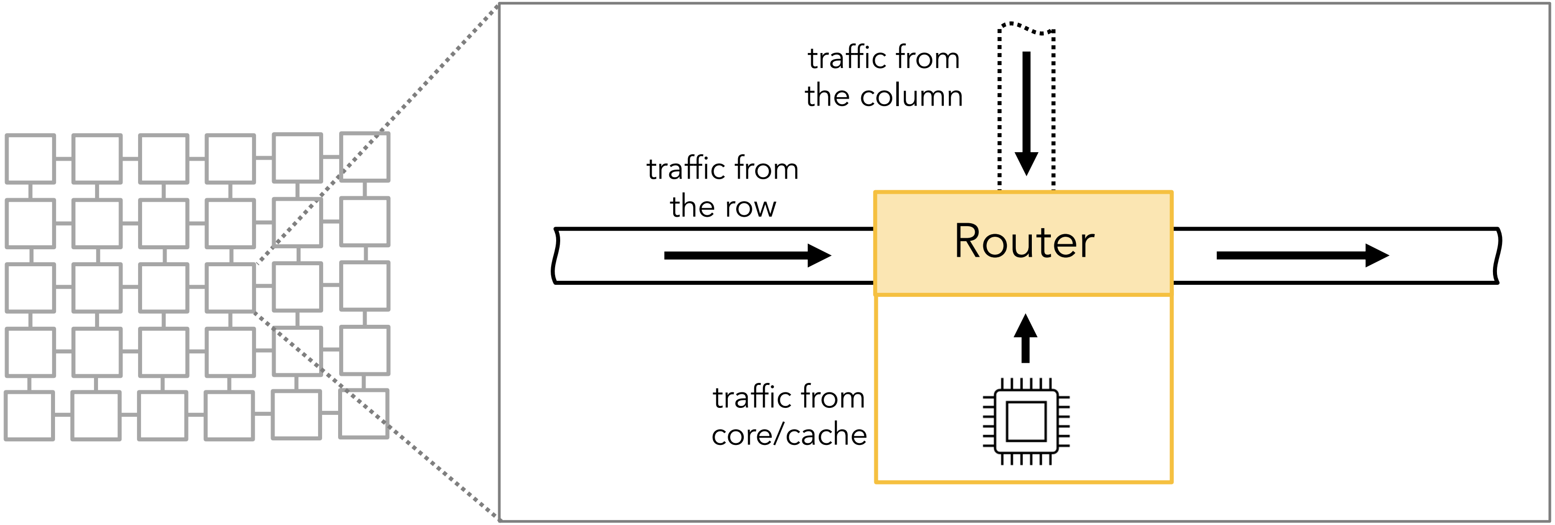


Contention

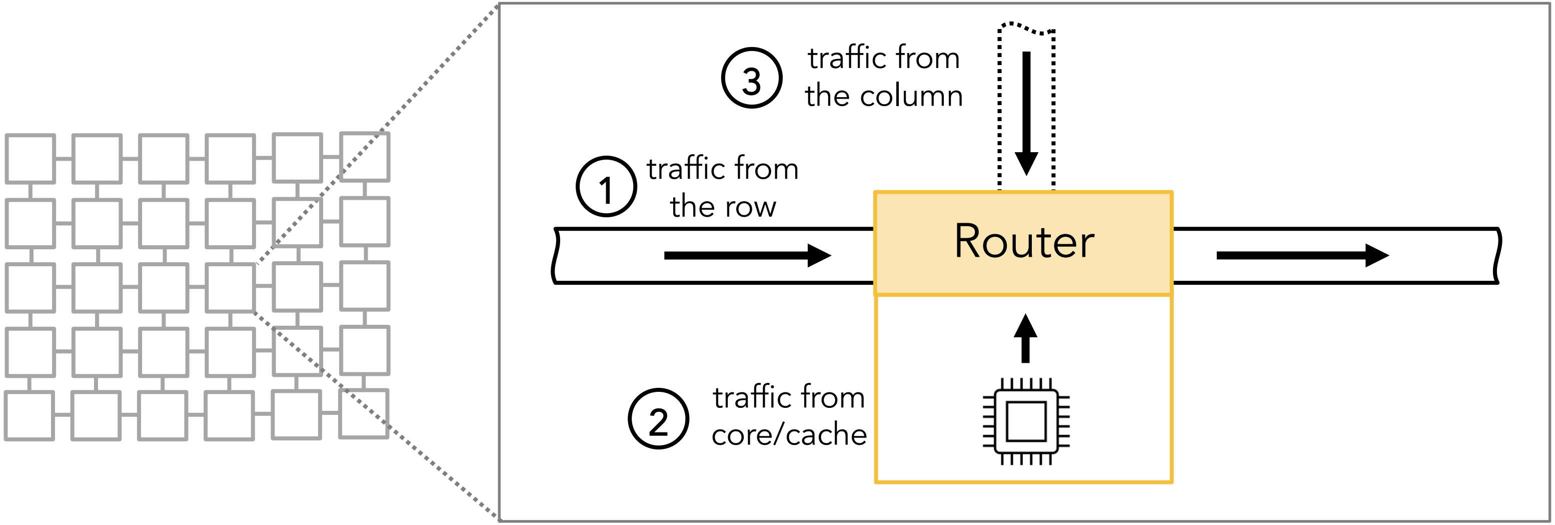


No contention

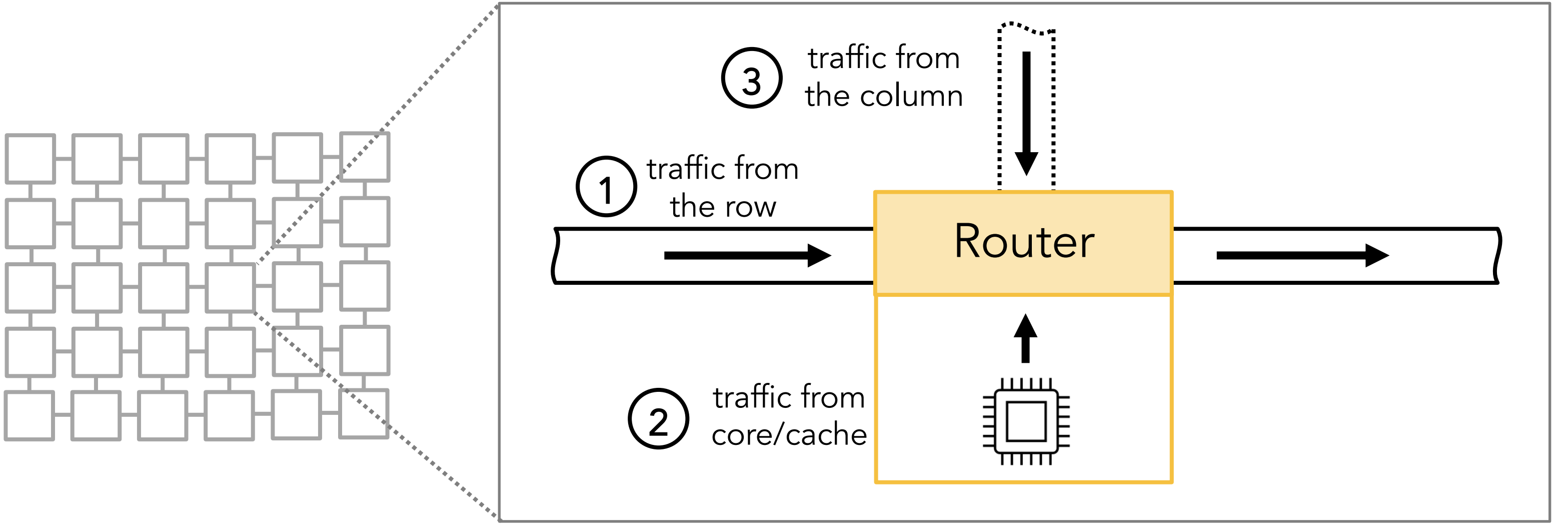
Priority Arbitration



Priority Arbitration

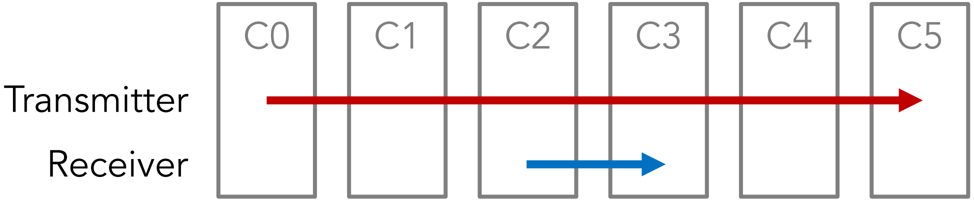


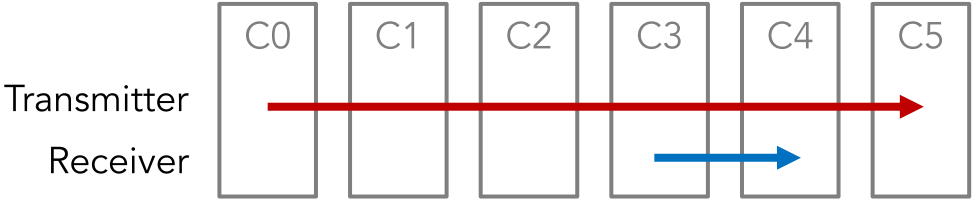
Priority Arbitration



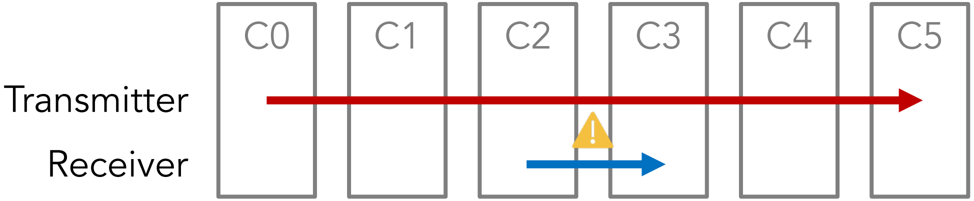
Transmitter traffic must have higher priority to delay receiver traffic.

Lane Scheduling

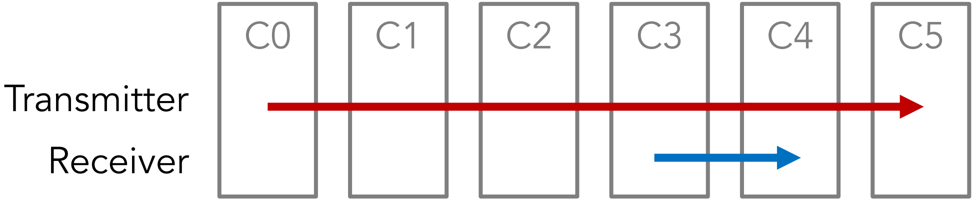




Lane Scheduling

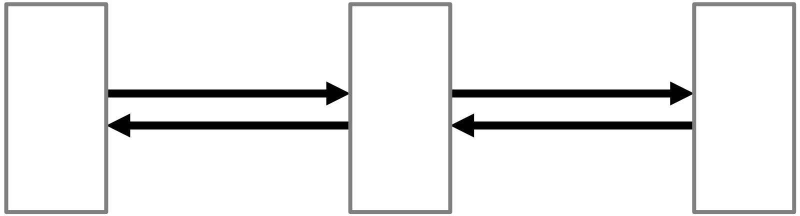


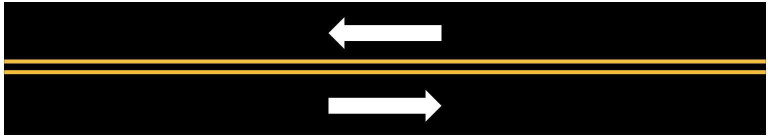
Contention



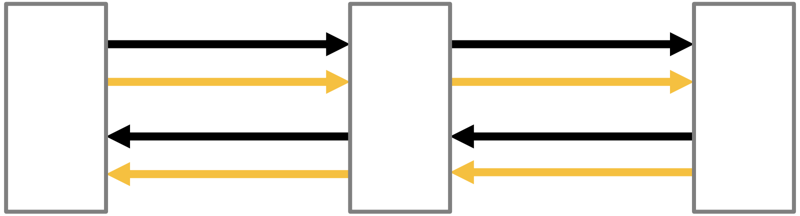
No contention

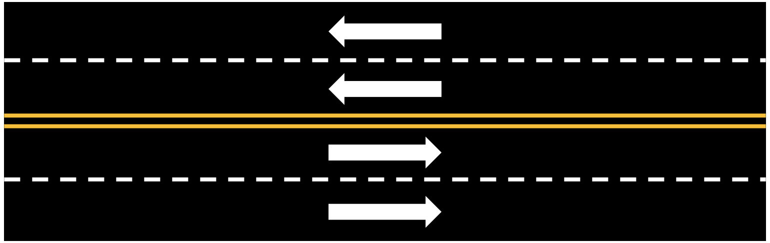
Lane Scheduling





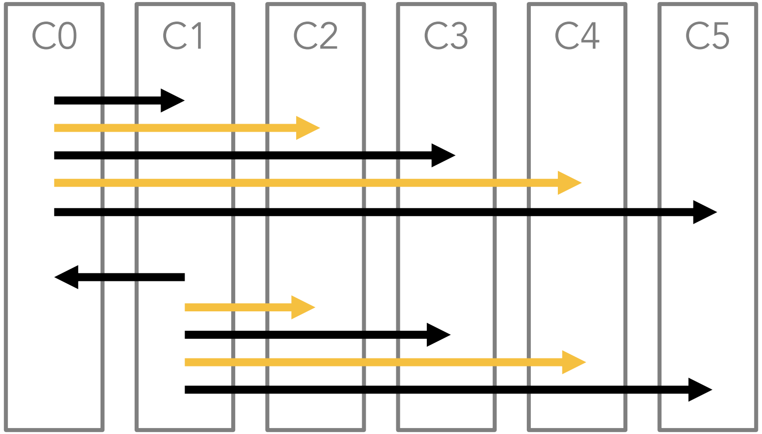
Lane Scheduling

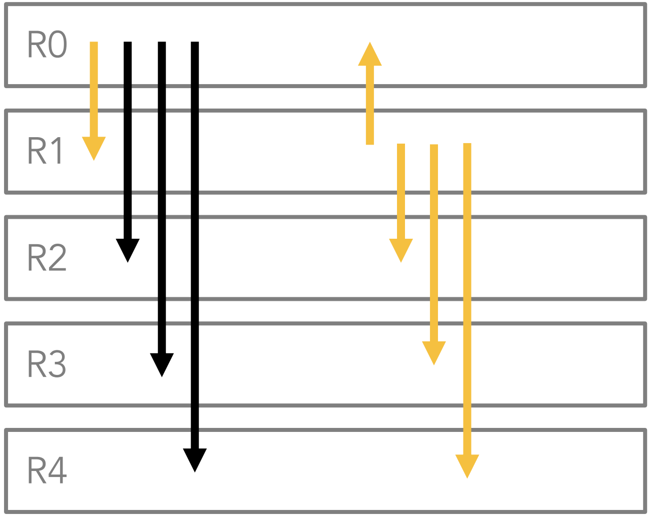




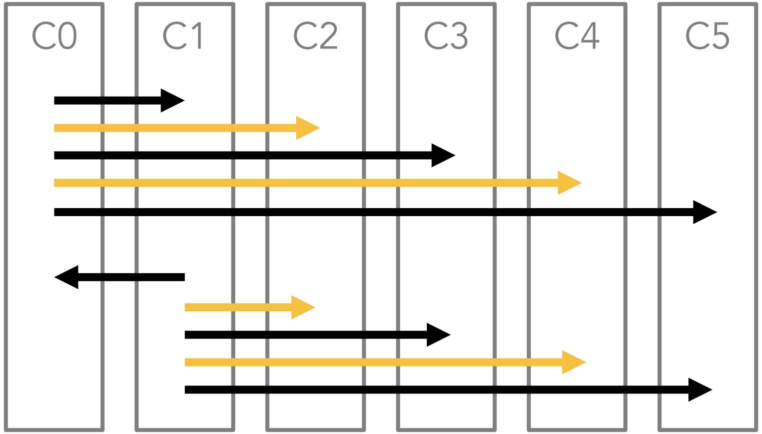
Two lanes per direction

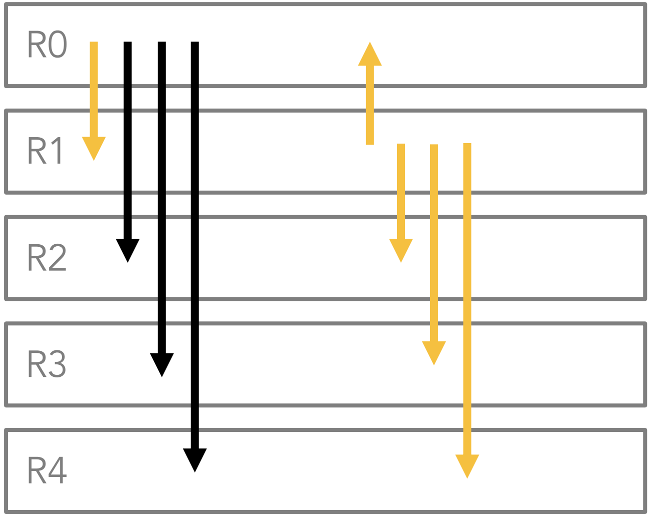
Lane Scheduling





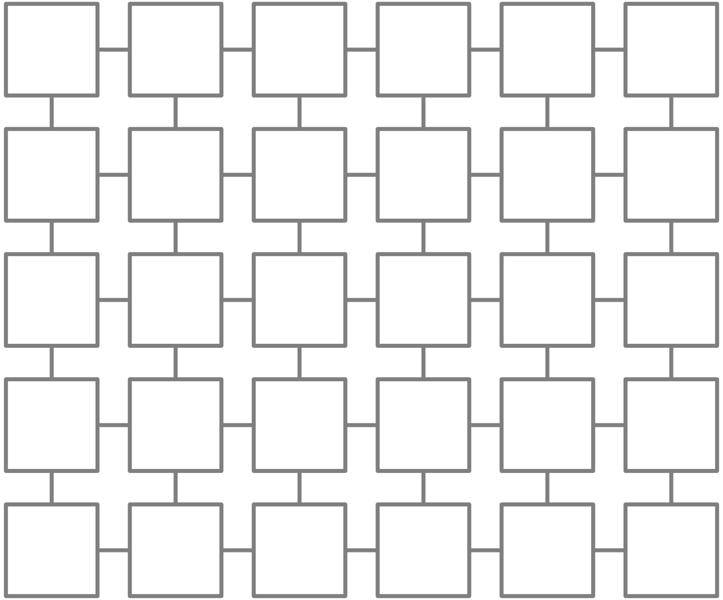
Lane Scheduling



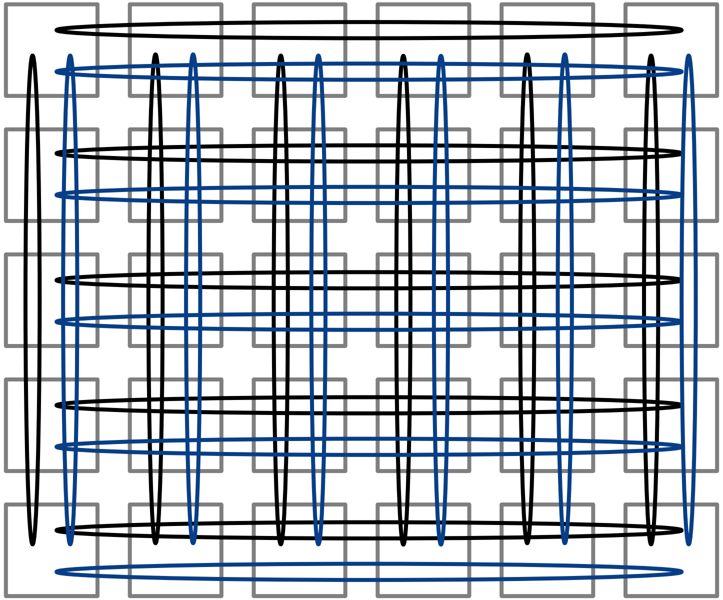


Traffic must travel on the same lane to contend.

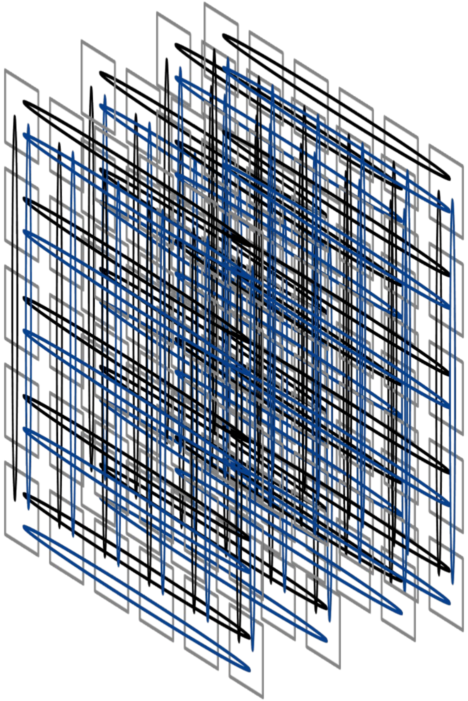
The Full Picture



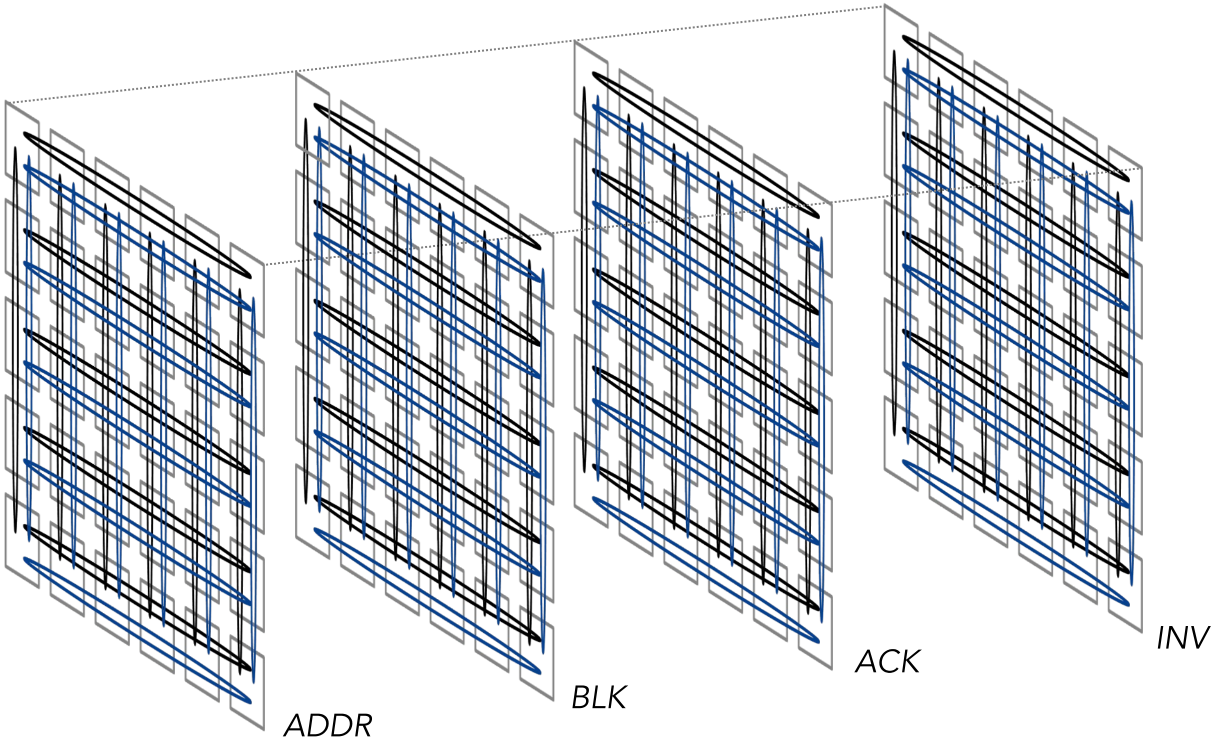
The Full Picture



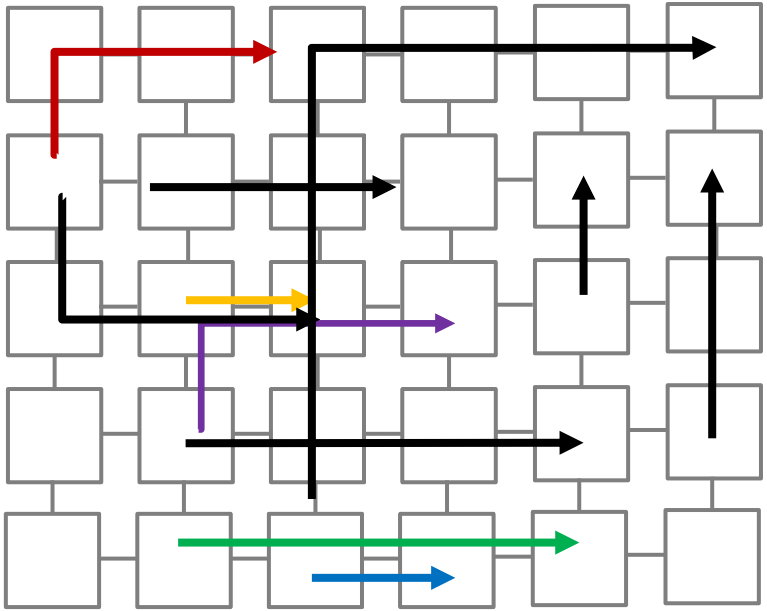
The Full Picture



The Full Picture



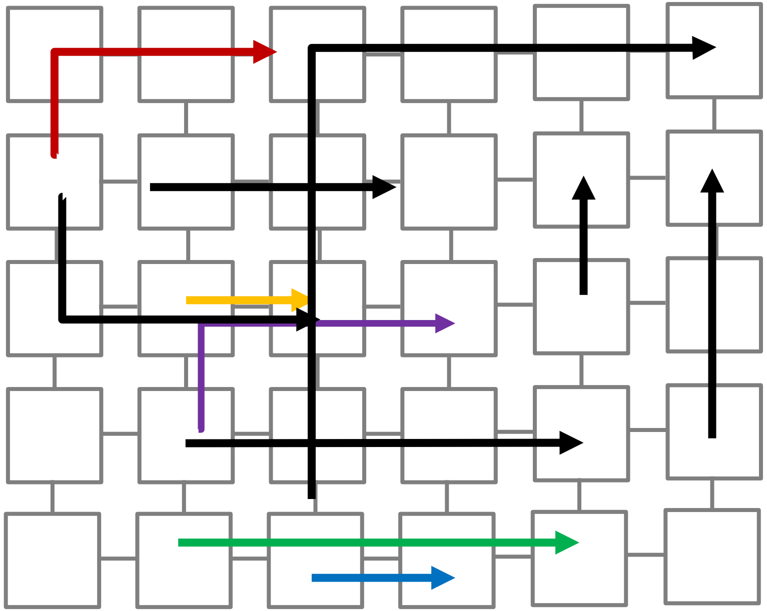
Research Questions



- Is it feasible to construct attacks by only exploiting contention on a mesh interconnect?

- Are there non-invasive approaches that can mitigate these attacks without requiring hardware modifications?

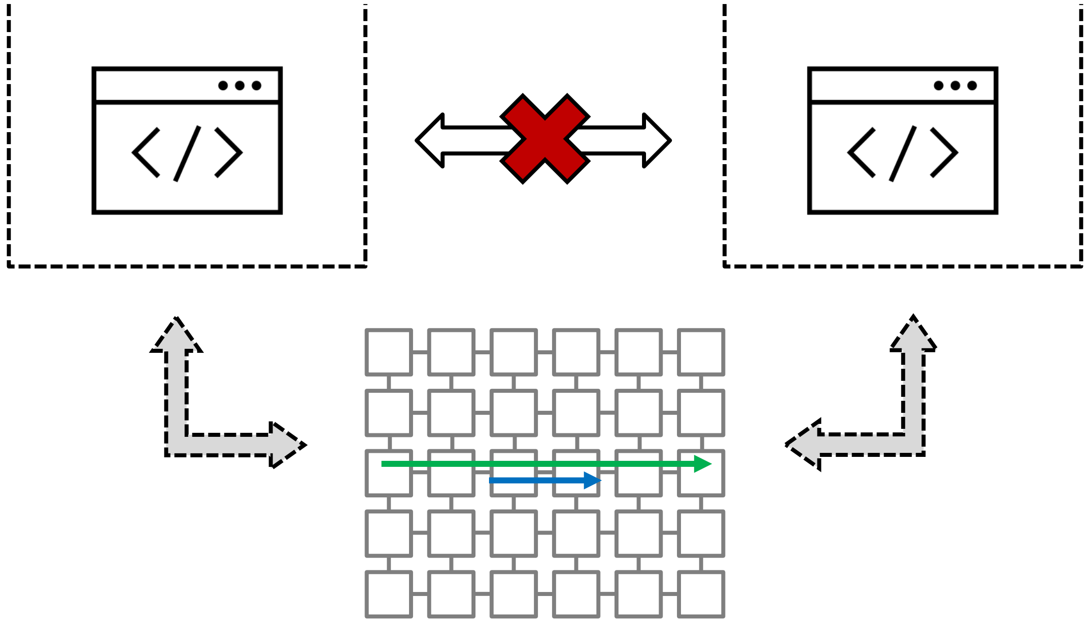
Research Questions



- Is it feasible to construct attacks by only exploiting contention on a mesh interconnect?

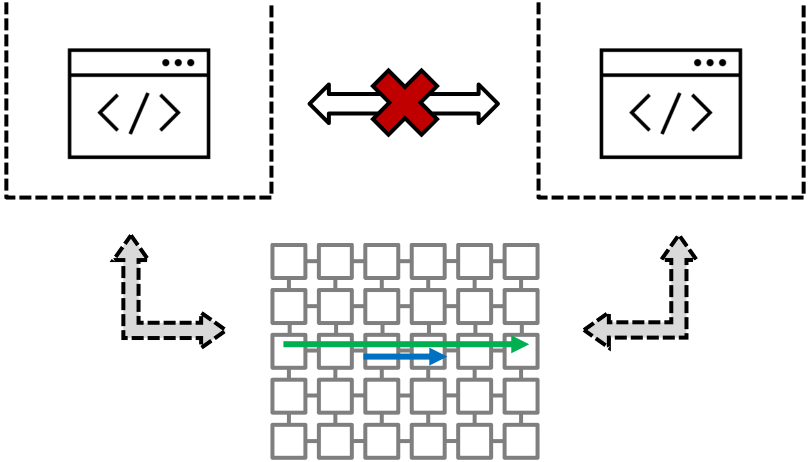
- Are there non-invasive approaches that can mitigate these attacks without requiring hardware modifications?

Covert Channel



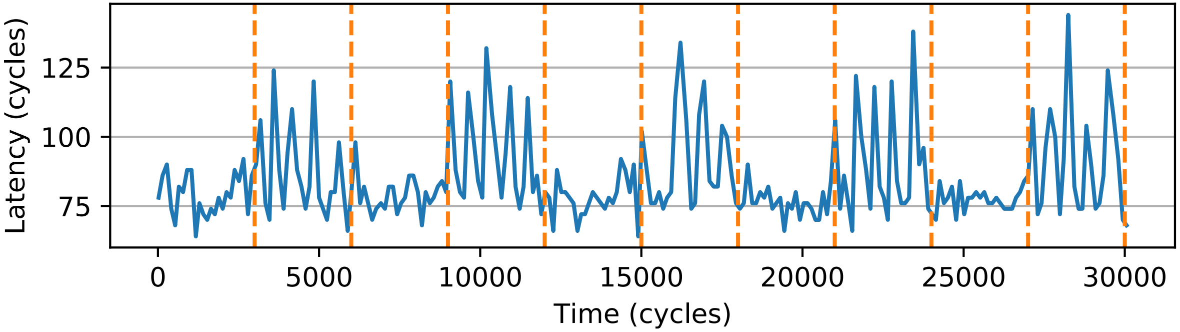
Covert Channel

- Transmit "1" → mesh contention
- Transmit "0" → idle



Covert Channel

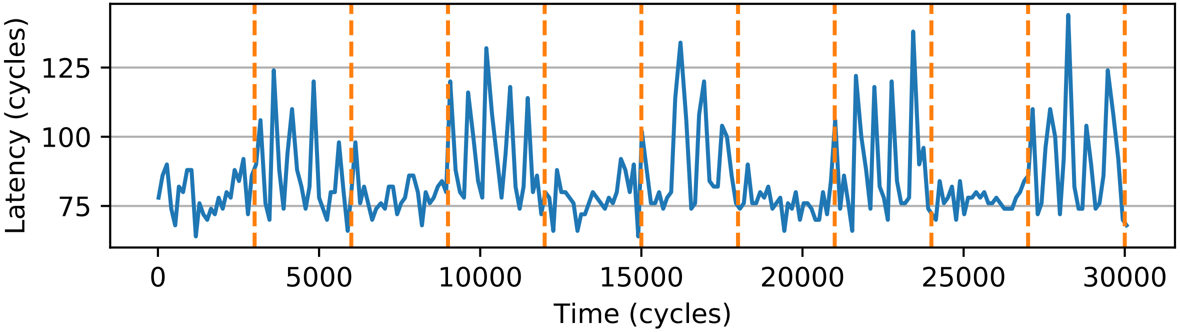
- Transmit “1” → mesh contention
- Transmit “0” → idle



Covert Channel

- Transmit “1” → mesh contention
- Transmit “0” → idle

Channel Capacity over 1.5 Mbps!

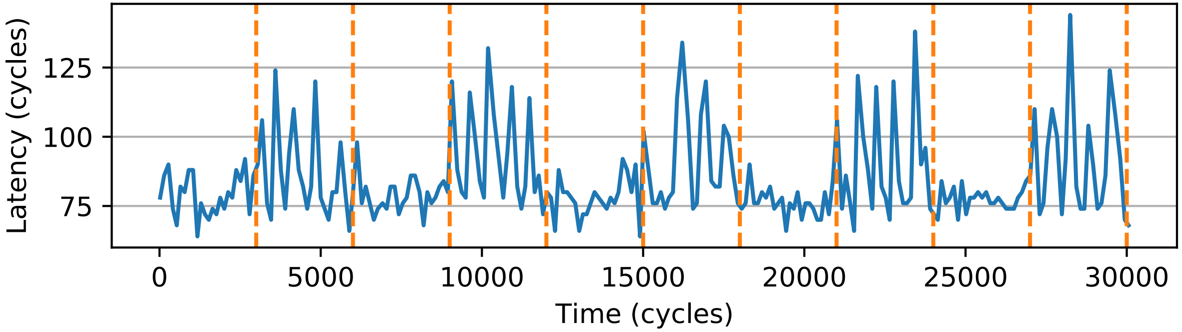


Covert Channel

- Transmit "1" → mesh contention
- Transmit "0" → idle

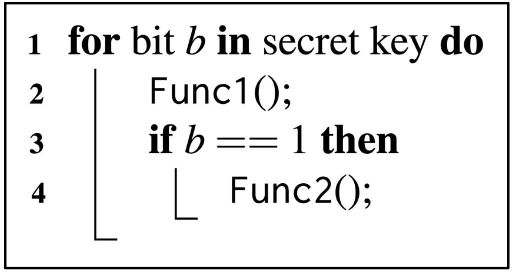
Channel Capacity over 1.5 Mbps!

Works across VMs!



Side-Channel Attack

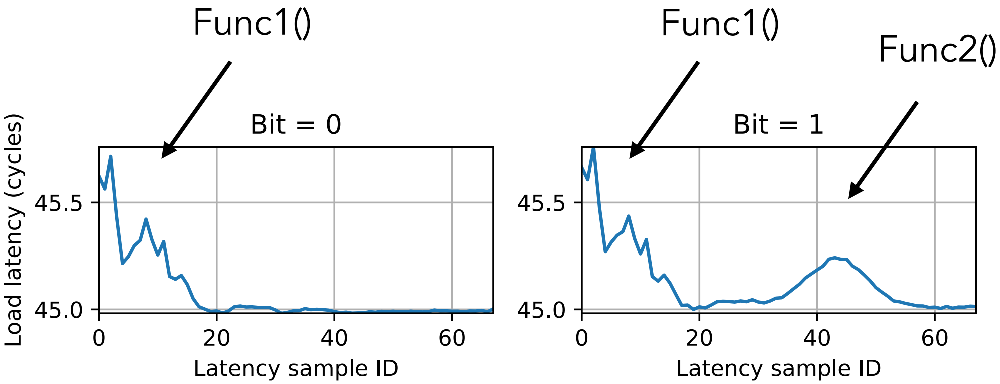
Side-Channel Attack



```
for bit $b$ in secret key do
    Func1();
    if $b == 1$ then
        Func2();
```

Used in vulnerable RSA and ECDSA implementations

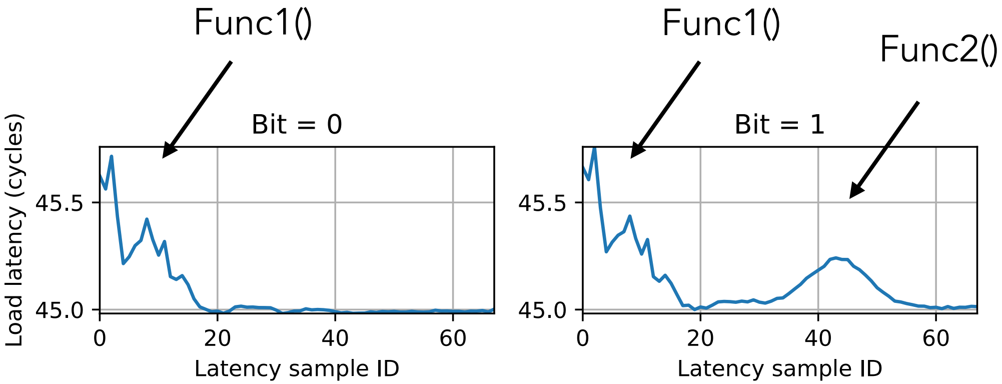
Side-Channel Attack



```
for bit $b$ in secret key do
    Func1();
    if $b == 1$ then
        Func2();
```

Used in vulnerable RSA and ECDSA implementations

Side-Channel Attack

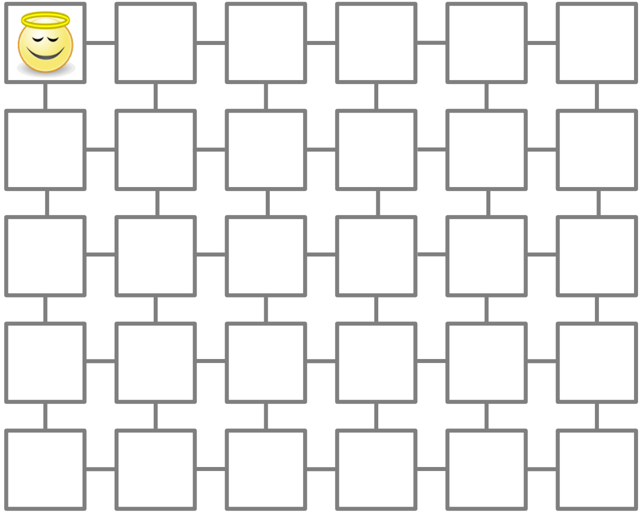


```
for bit $b$ in secret key do
    Func1();
    if $b == 1$ then
        Func2();
```

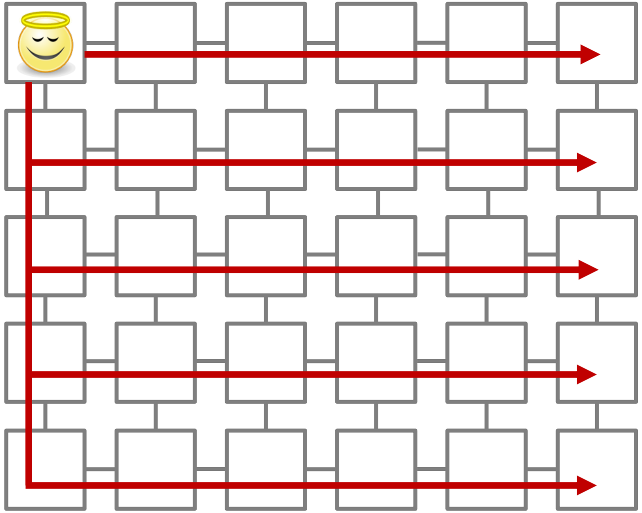
Used in vulnerable RSA and ECDSA implementations

More details in the paper

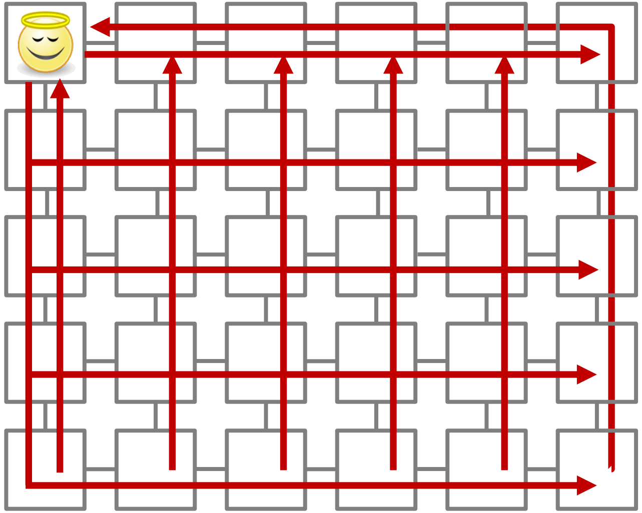
Impact of the Attacker's Placement



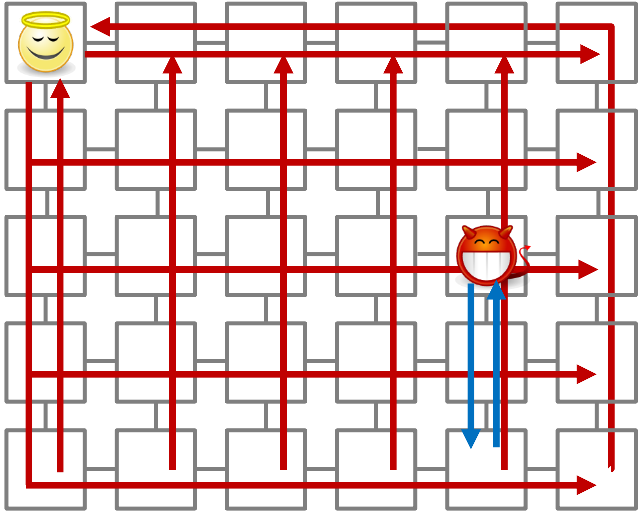
Impact of the Attacker’s Placement



Impact of the Attacker’s Placement

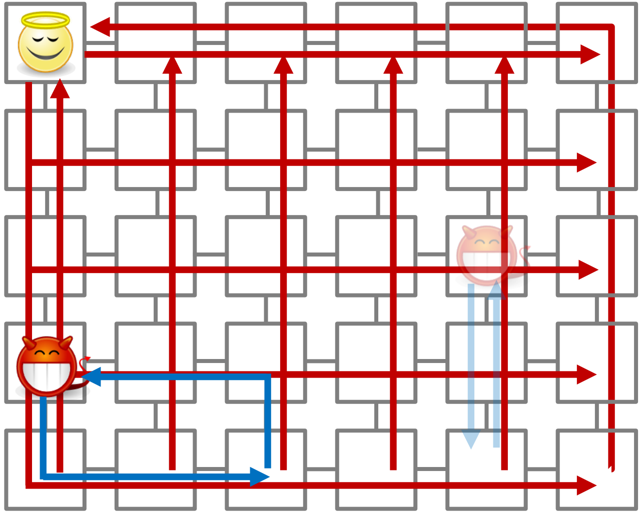


Impact of the Attacker's Placement



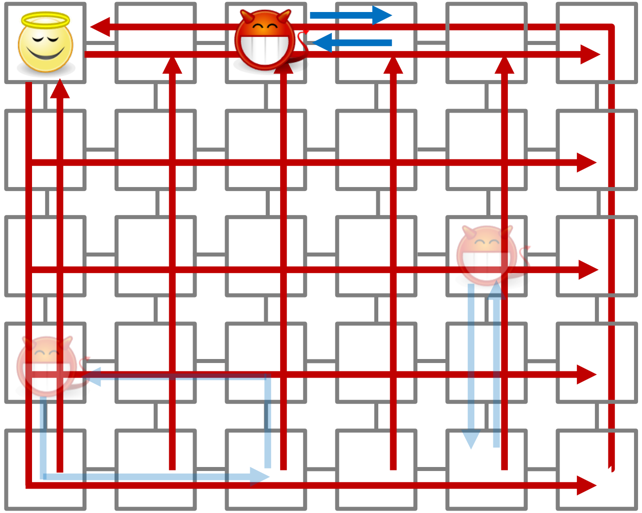
- Best attacker placement?

Impact of the Attacker’s Placement



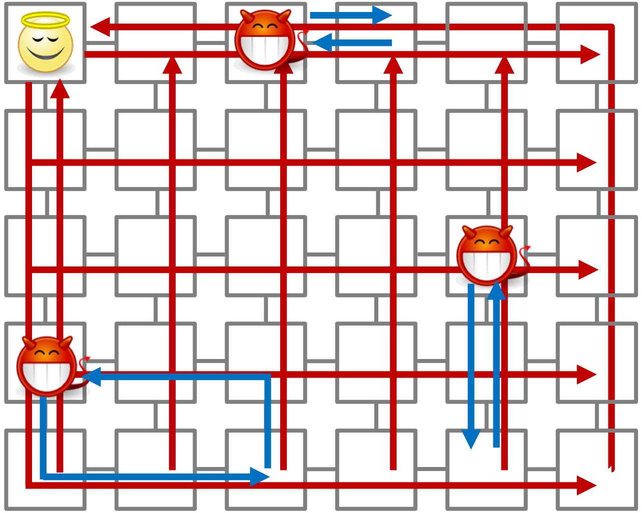
- Best attacker placement?

Impact of the Attacker’s Placement



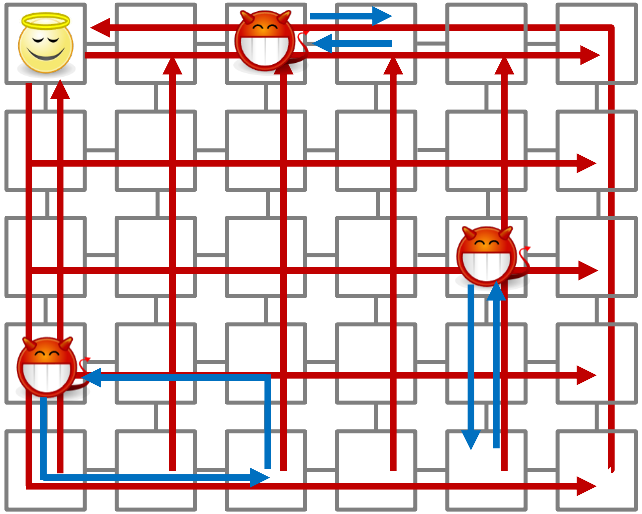
- Best attacker placement?

Impact of the Attacker’s Placement



- Best attacker placement?
- 23 cores * 25 slices = 575 attacker placement options!

Impact of the Attacker’s Placement



- Best attacker placement?
- 23 cores * 25 slices = 575 attacker placement options!

We construct an *analytical model* to rank placements

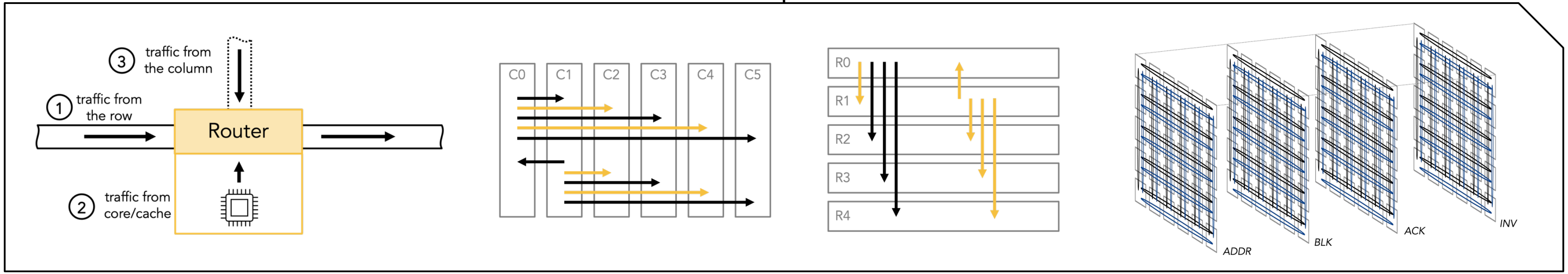
Impact of the Attacker’s Placement

Attacker’s placement

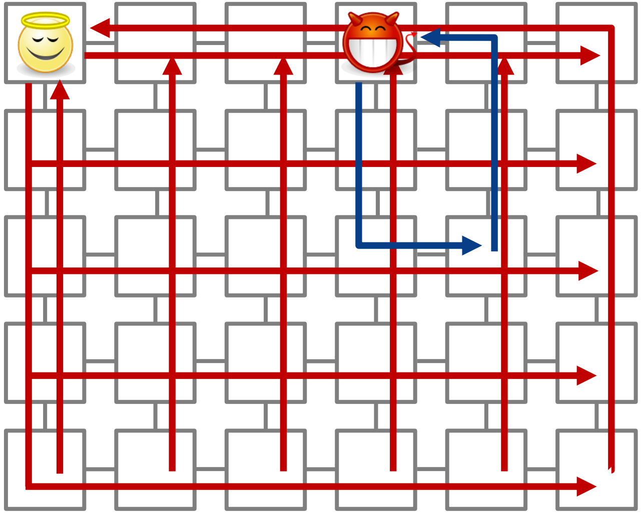
Victim’s core

Analytical model

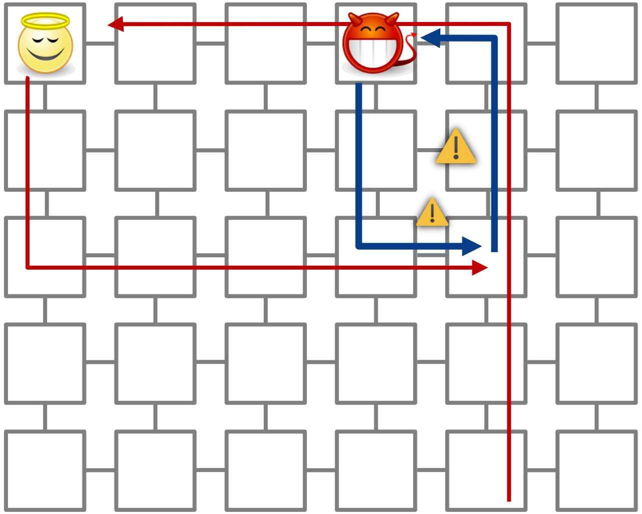
Score from 0 to 10 (*higher = more contention)*



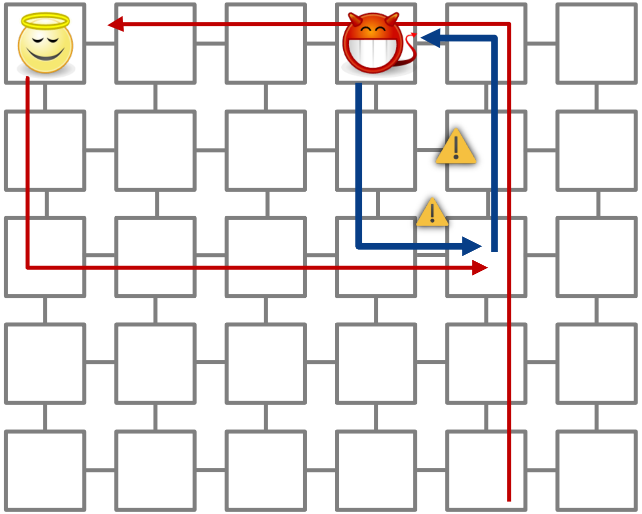
Impact of the Attacker's Placement



Impact of the Attacker’s Placement

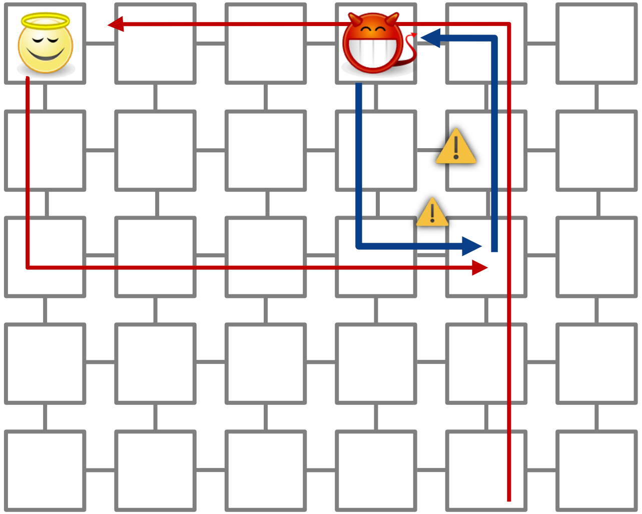


Impact of the Attacker’s Placement

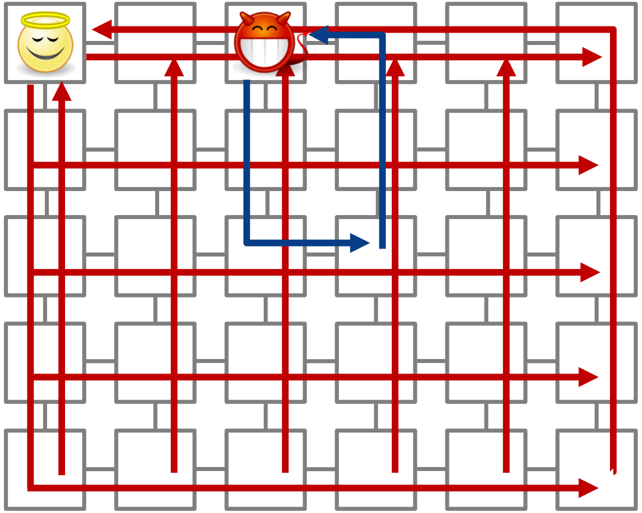


Score = 1

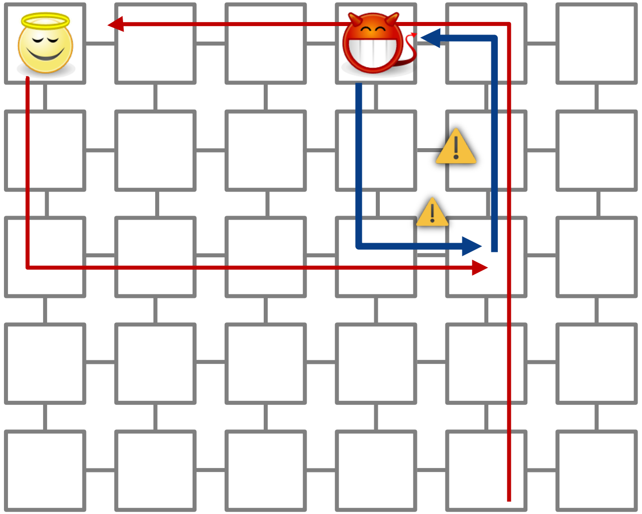
Impact of the Attacker’s Placement

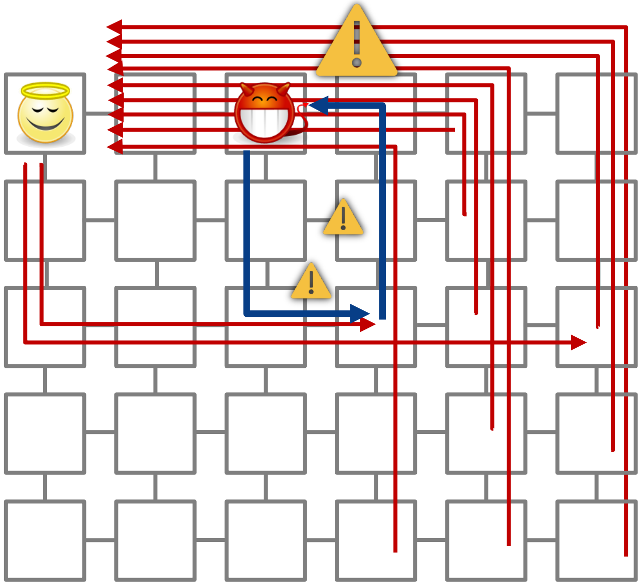


Score = 1



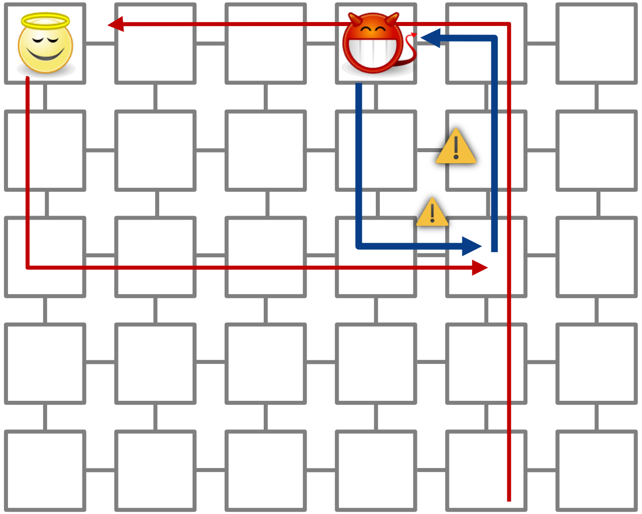
Impact of the Attacker’s Placement



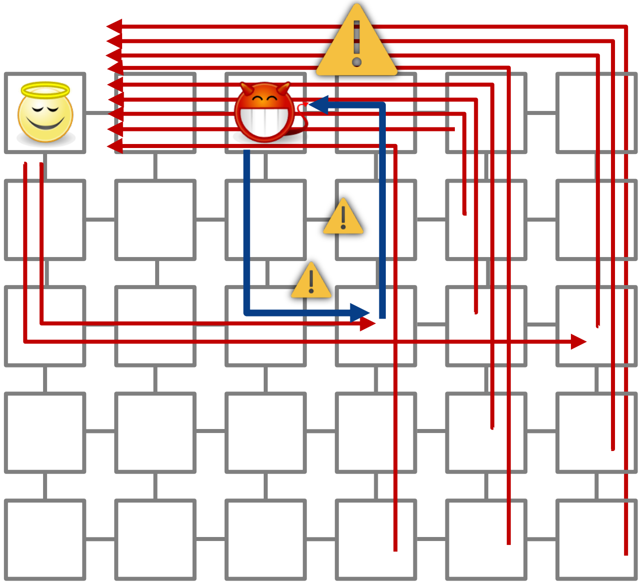


Score = 1

Impact of the Attacker’s Placement

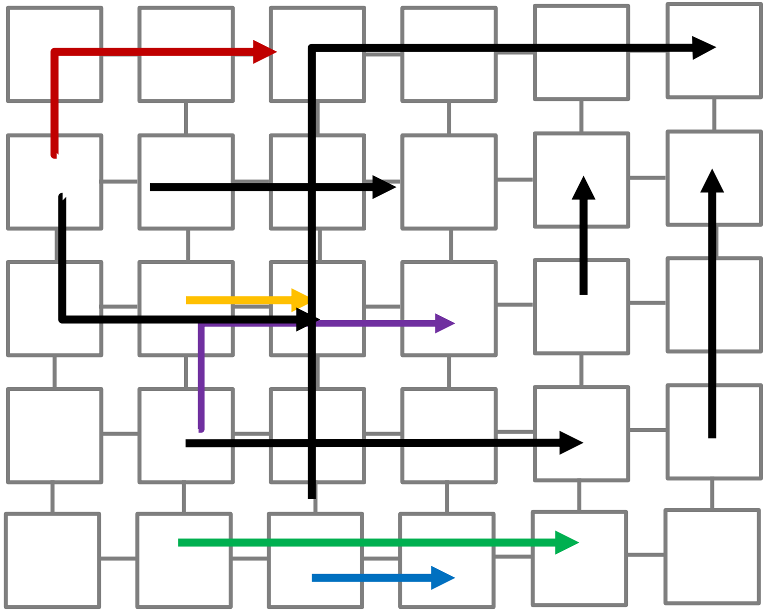


Score = 1



Score = 10

Research Questions

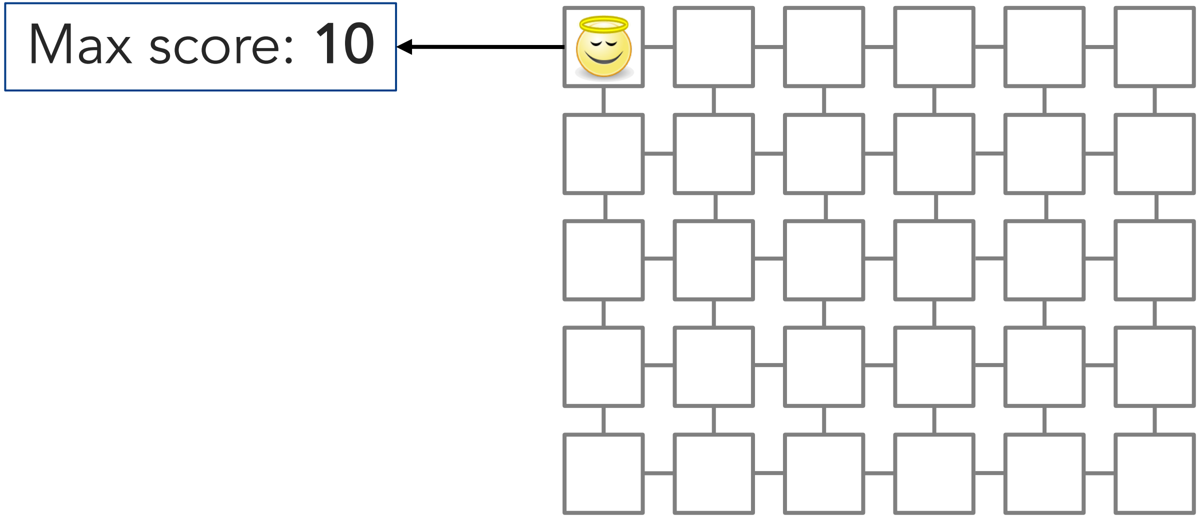


- Is it feasible to construct attacks by only exploiting contention on a mesh interconnect?

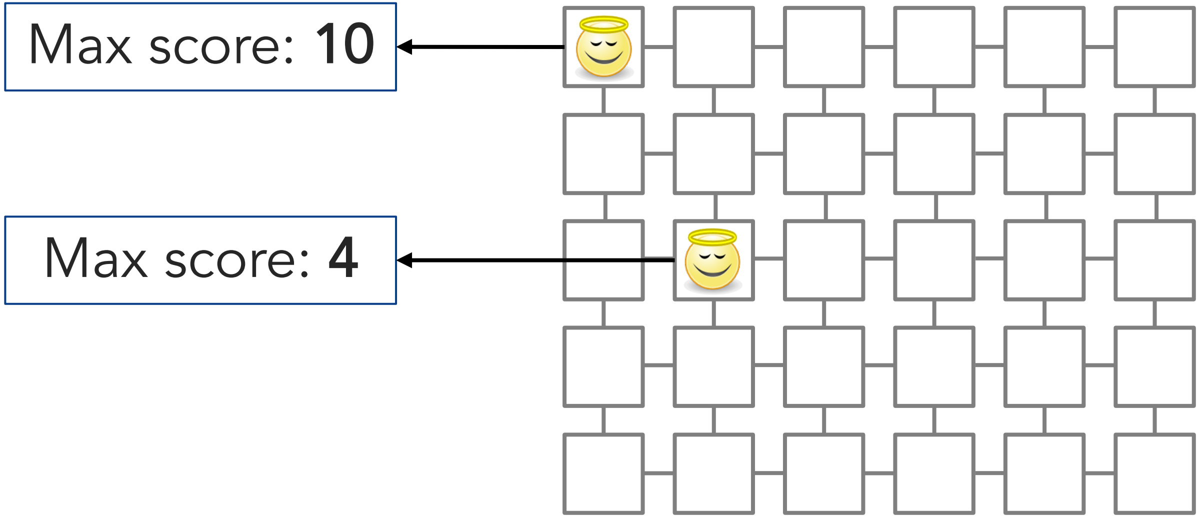
- Are there non-invasive approaches that can mitigate these attacks without requiring hardware modifications?

Impact of the Victim's Core

Impact of the Victim's Core



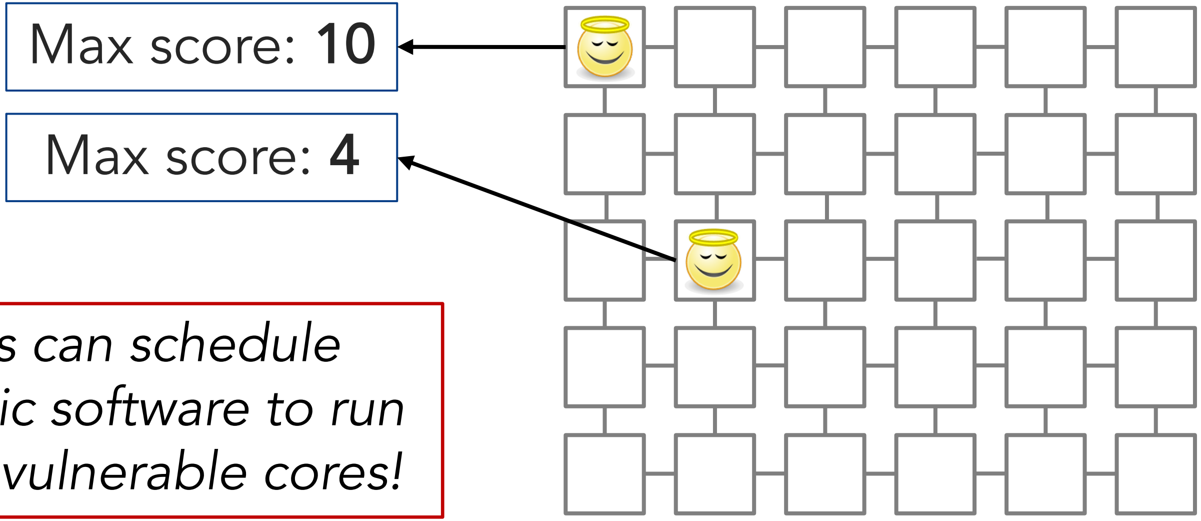
Impact of the Victim’s Core



Max score: **10**

Max score: **4**

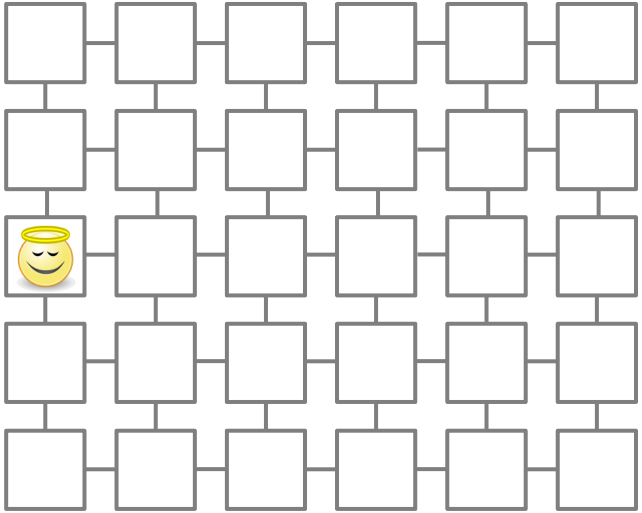
Mitigation Insight #1



Defenders can schedule cryptographic software to run on the least vulnerable cores!

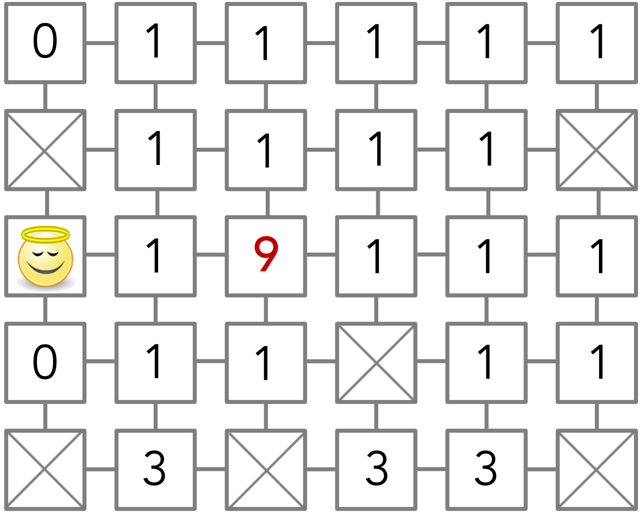
Impact of the Attacker’s Core

Max score: **?**



Impact of the Attacker’s Core

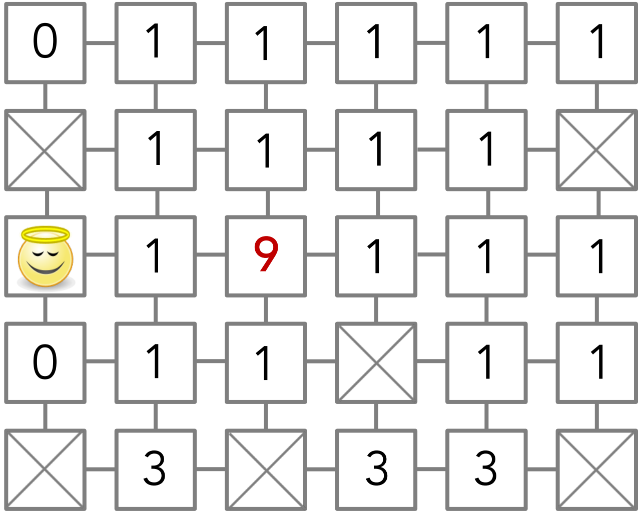
Max score: 9



0	1	1	1	1	1
☒	1	1	1	1	☒
😇	1	9	1	1	1
0	1	1	☒	1	1
☒	3	☒	3	3	☒

Impact of the Attacker’s Core

Max score: 9



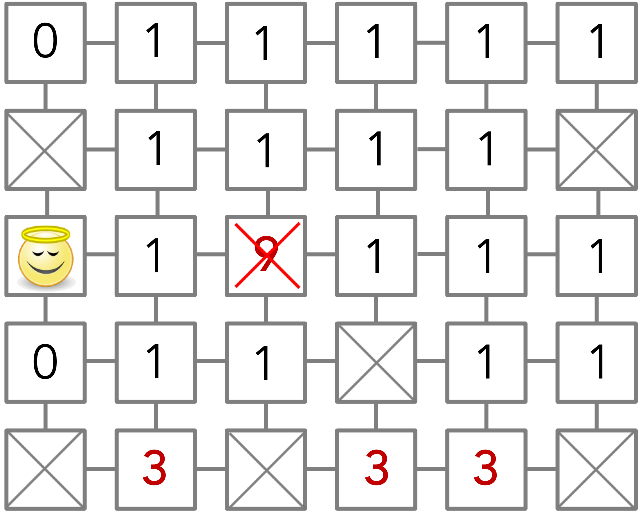
0	1	1	1	1	1
☒	1	1	1	1	☒
😇	1	9	1	1	1
0	1	1	☒	1	1
☒	3	☒	3	3	☒

Can we prevent the attacker from taking good placements?

Impact of the Attacker’s Core

~~Max score: 9~~

Max score: **3**



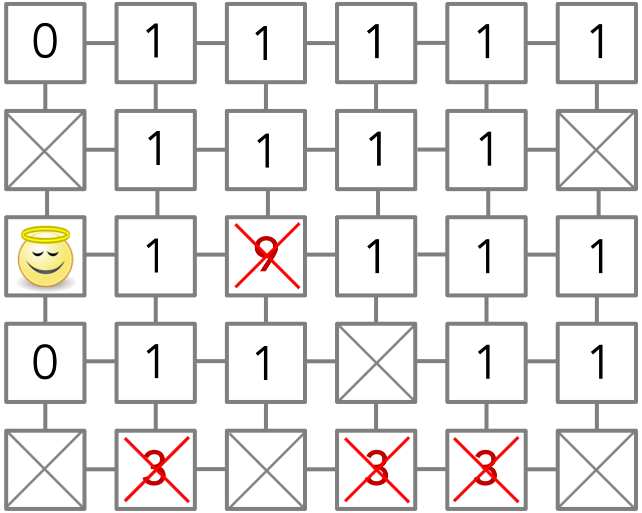
0	1	1	1	1	1
☒	1	1	1	1	☒
😇	1	~~9~~	1	1	1
0	1	1	☒	1	1
☒	3	☒	3	3	☒

Impact of the Attacker’s Core

~~Max score:~~ 9

~~Max score:~~ 3

Max score: 1



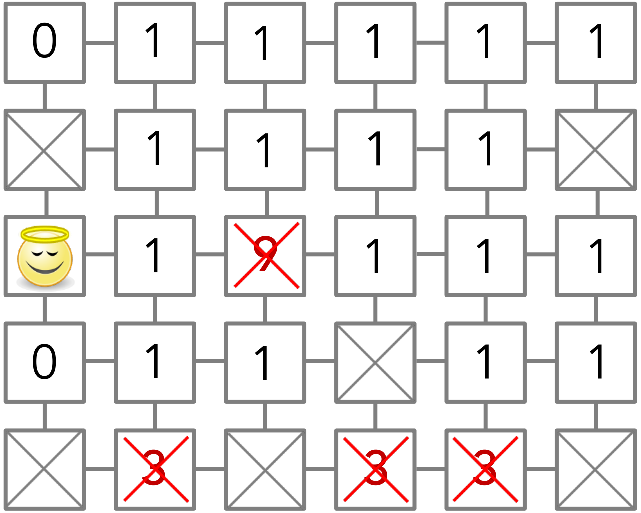
0	1	1	1	1	1
☒	1	1	1	1	☒
😇	1	~~9~~	1	1	1
0	1	1	☒	1	1
☒	~~3~~	☒	~~3~~	~~3~~	☒

Mitigation Insight #2

~~Max score~~: 9

~~Max score~~: 3

Max score: 1



0	1	1	1	1	1
☒	1	1	1	1	☒
😇	1	~~9~~	1	1	1
0	1	1	☒	1	1
☒	~~3~~	☒	~~3~~	~~3~~	☒

Defenders can reserve certain cores for the victim’s security domain!

Conclusion

- On-chip interconnects remain an overlooked microarchitectural attack surface, ignored by existing “domain isolation” defenses.
- This work demonstrates the feasibility of side channel attacks on the mesh interconnect.
- This work offers new insights into mitigating these attacks without changing the hardware.

https://github.com/CSAIL-Arch-Sec/dont-mesh-around

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED