Optimizing Power and Performance for Reliable On-Chip Networks

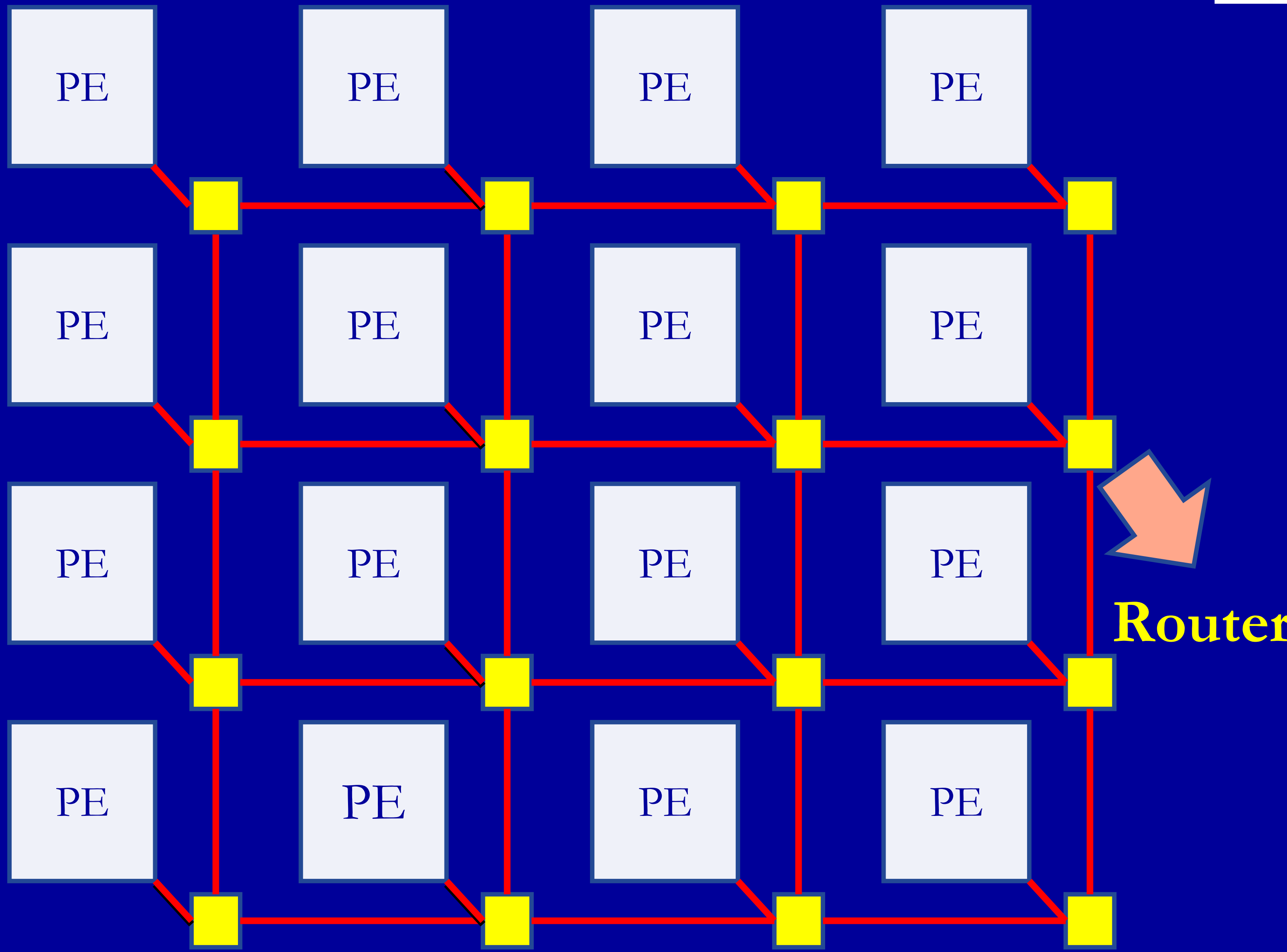
Aditya Yanamandra, Soumya Eachempati, Niranjan Soundararajan, Vijaykrishnan Narayanan,
Mary Jane Irwin, Ramakrishnan Krishnan

The Pennsylvania State University

Overview

- Motivation
- Contributions
- Analytical model
- Implementation details
- Conclusion

CMP Layout



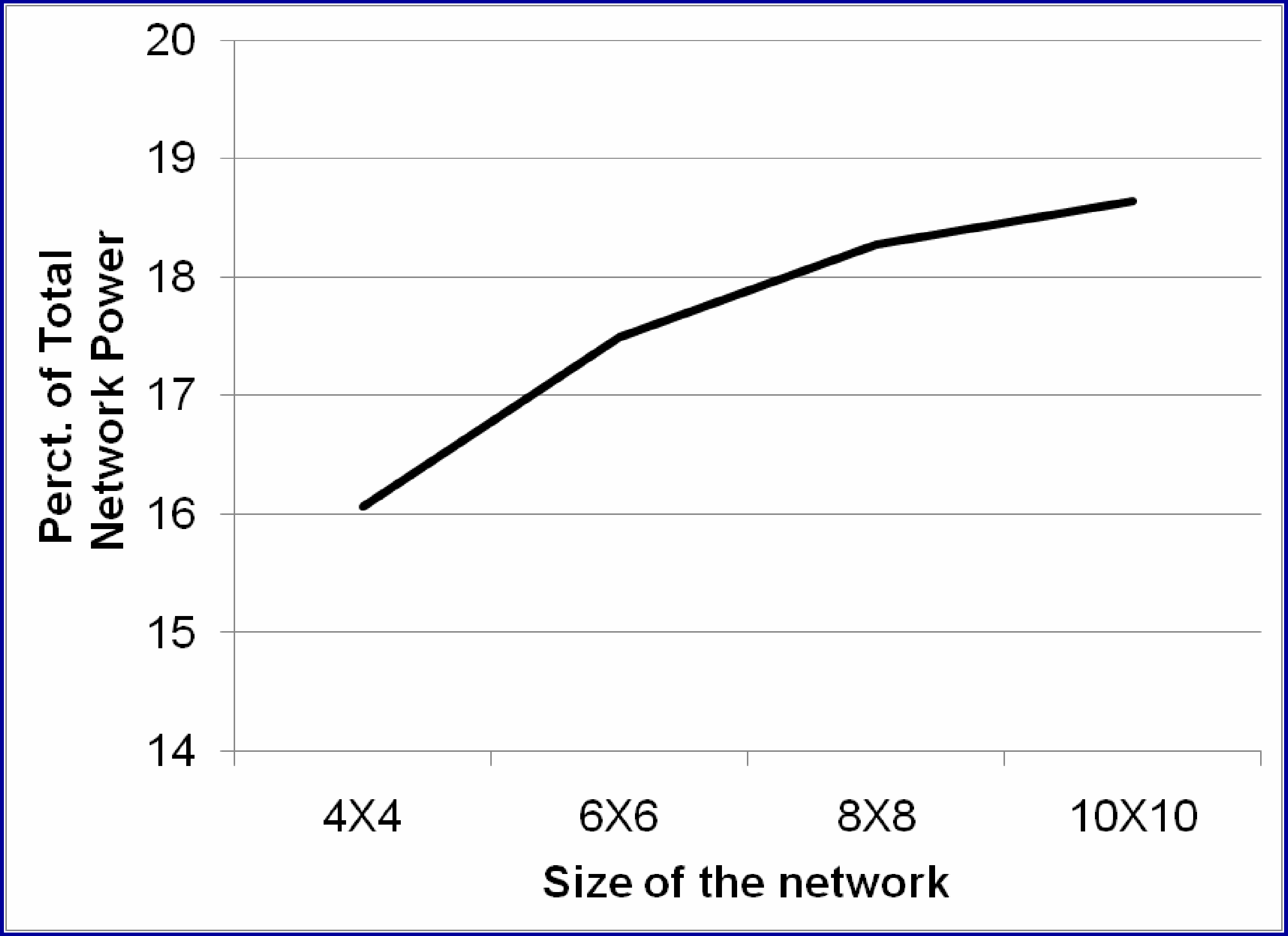
Motivation

- NOC power is a key limiter for adoption in industry.
- NOCs in modern chips need data protection from soft errors.
- Error protection takes power and is not needed if errors can be tolerated.

Router Pipeline

<table>
<tr><td>ECC (Header)</td><td>ECC (body)</td><td rowspan="3">ST</td><td rowspan="3">LT</td></tr>
<tr><td>RC</td><td rowspan="2">SA</td></tr>
<tr><td>VA</td></tr>
</table>

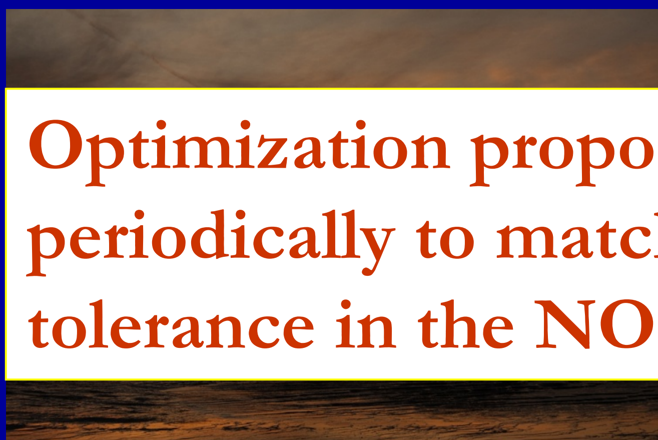
ECC power graph

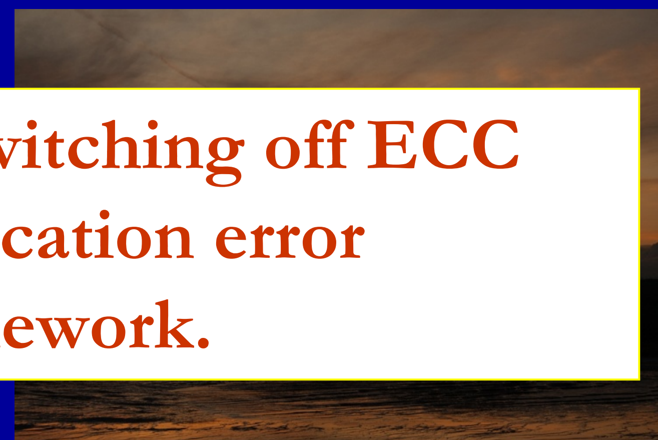


Turning off ECC partially

Errors can be tolerated

Spot the 10 pixels that are different!!!





Optimization proposed: Switching off ECC periodically to match application error tolerance in the NOC framework.

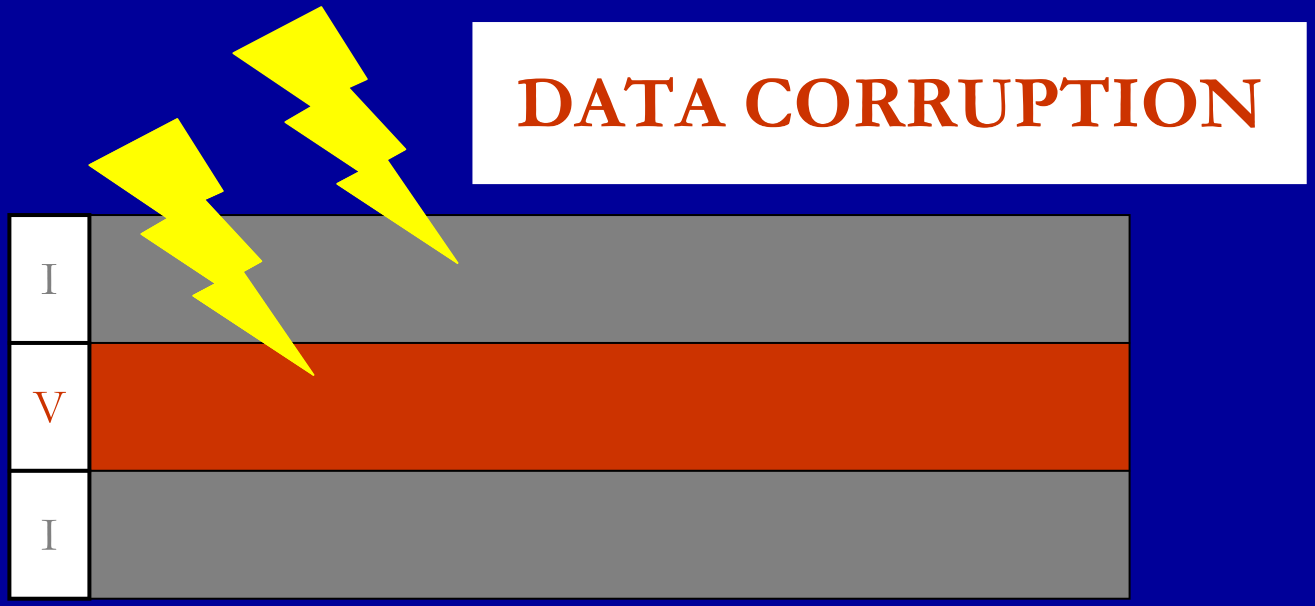
Contributions

- Modeling target reliability
  - NOC buffer usage
  - Perct. time error protection is switched on.
- Implications of power and performance for this model.
- Three implementations
  - Static SAVF
  - Static MAVF
  - Dynamic MAVF

Overview

- Motivation
- Contributions
- **Analytical model**
- Implementation details
- Conclusion

Architectural Vulnerability Factor



Not all SEU's lead to errors

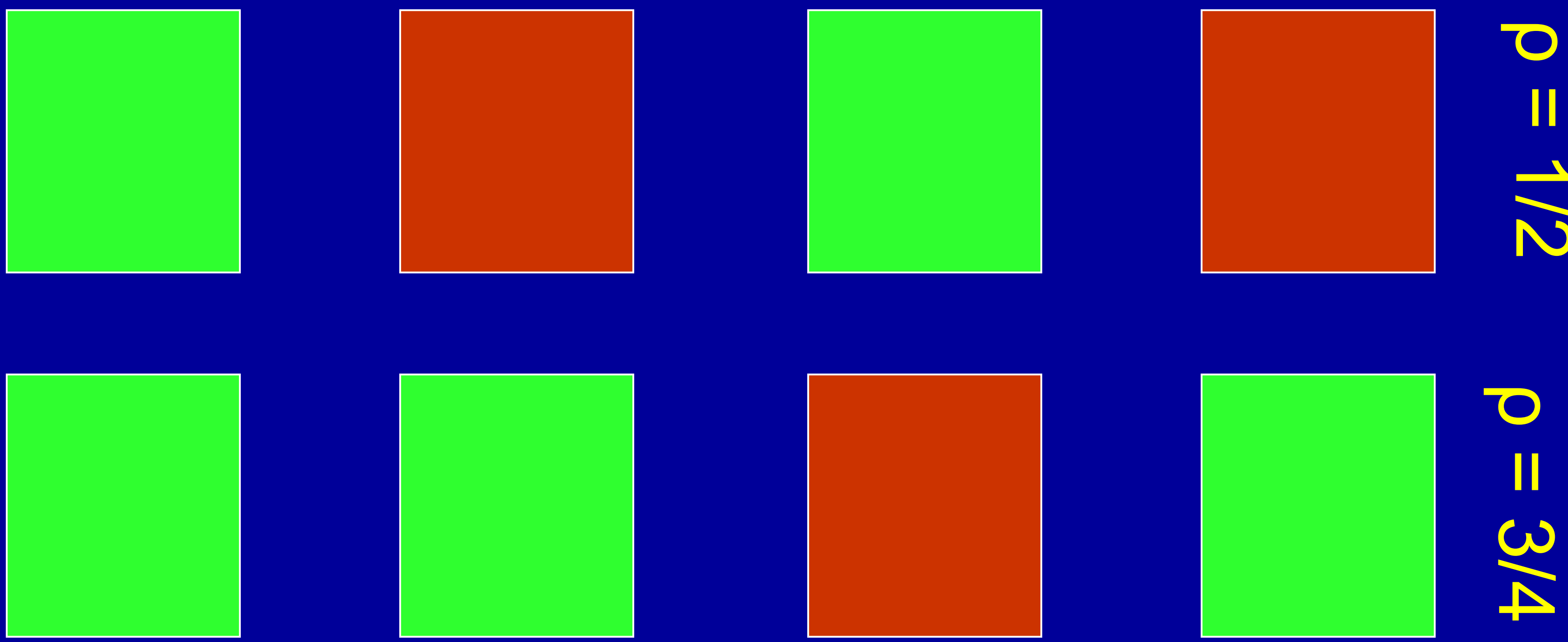
$$\lambda_{eff} = AVF * \lambda_{raw}$$

Duty Cycle (ρ)

- Fraction of time ECC is turned on.
- From the previous example

ECC ON

ECC OFF



Residual Error Rate

$$\lambda_{res} = (1 - \rho * c)\lambda_{eff}$$

- c – coverage of the ECC.
- ρ – duty cycle of ECC

Thus, we get

$$\lambda_{res} = (1 - \rho * c) * AVF * \lambda_{raw}$$

Error tolerance metric (k)

- Define **'*k*'** (error tolerance) ratio of residual error rate to raw error rate
- $k = \lambda_{res} / \lambda_{raw}$
- For Single Event Upsets with Single Error Correction (thus c = 1), we get

$$\rho = 1 - \frac{k}{AVF}$$

Experimental Setup

- In-house cycle accurate NOC simulator.
- Router power from Orion
- 32 bit Encoder and decoder: Synopsys Design Compiler

	Encoder	Decoder
Leakage	55nW	493nW
Dynamic	04.mW	4mW

- Base case: ECC turned on 100% of the time.
- Applications from PARSEC suite.

Overview

- Motivation
- Contributions
- Analytical model
- **Implementation details and results**
  - SAVF results
  - MAVF
- Conclusion

Guaranteeing Reliability

- By turning off ECC for (1-ρ) fraction of the time, we still achieve the desired reliability guarantees.
- Power savings is (1-ρ)% = k/AVF
- Thus, power savings can be increased the bounding the AVF value below one.

Bounding AVF

- Is done by bounding buffer utilization.

✓ Pro: Gives higher power savings.

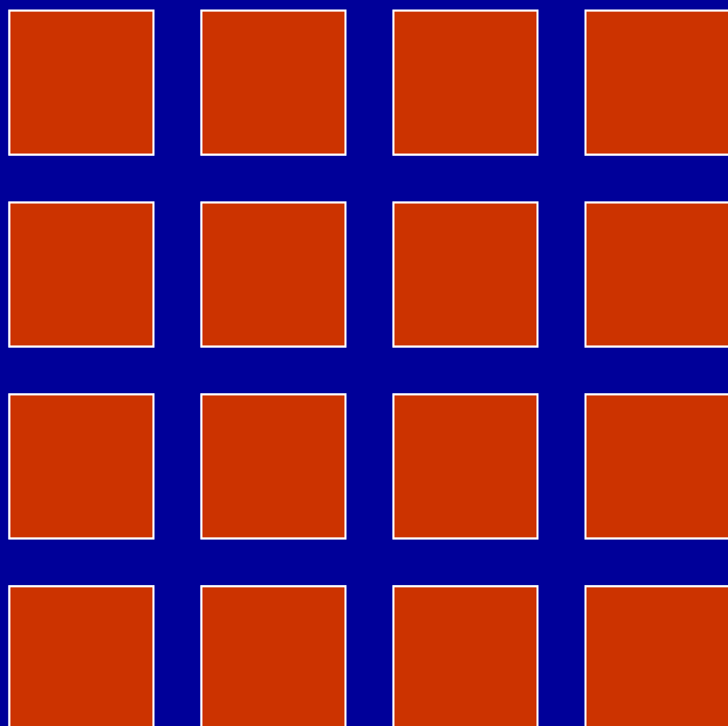
X Con: Lower throughput

- Single AVF domain. (SAVF)
  - All routers have same AVF bound
- Multiple AVF domains (MAVF)
  - Different bounds for each router.

Overview

- Motivation
- Contributions
- Analytical model
- **Implementation details and results**
  - **SAVF**
  - MAVF
- Conclusion

Understanding SAVF



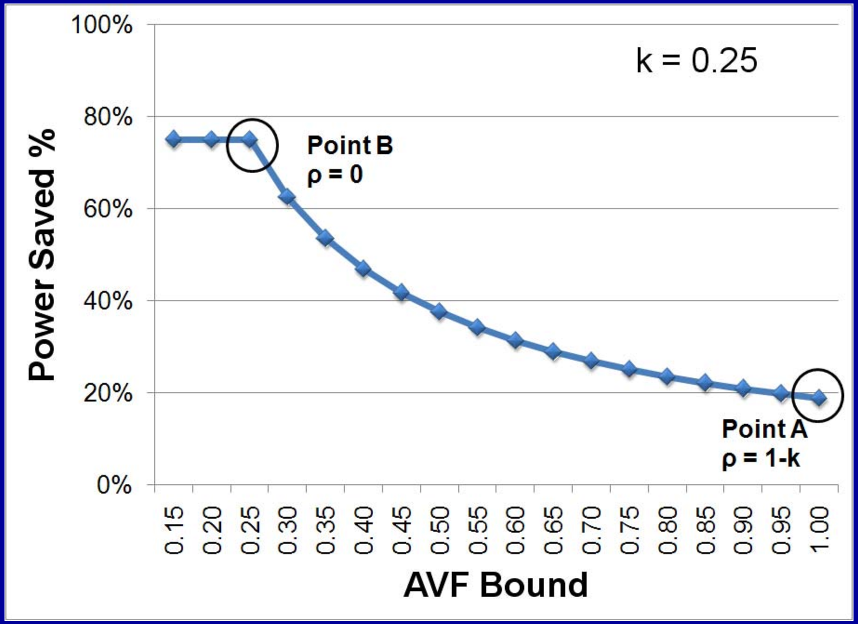
CYCLE 1

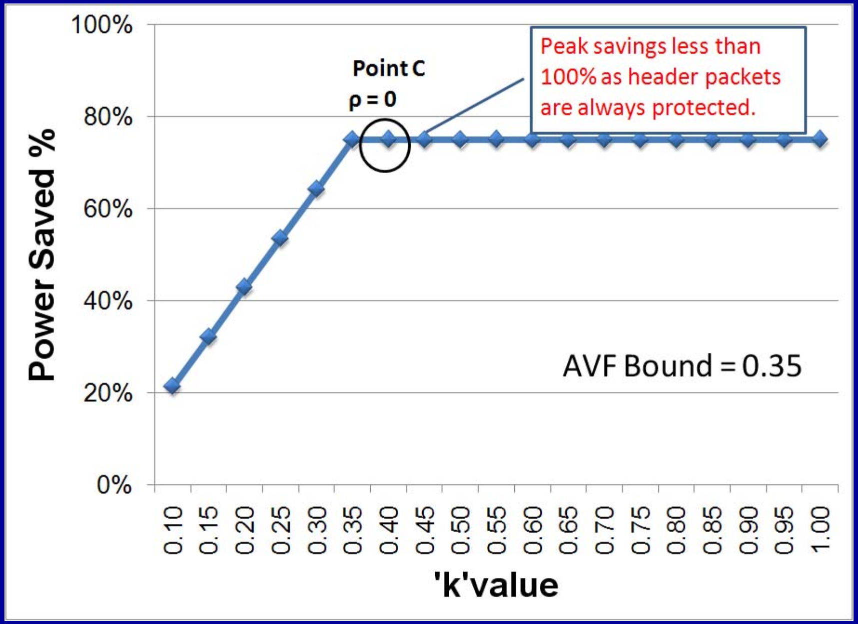
CYCLE 2

CYCLE 3

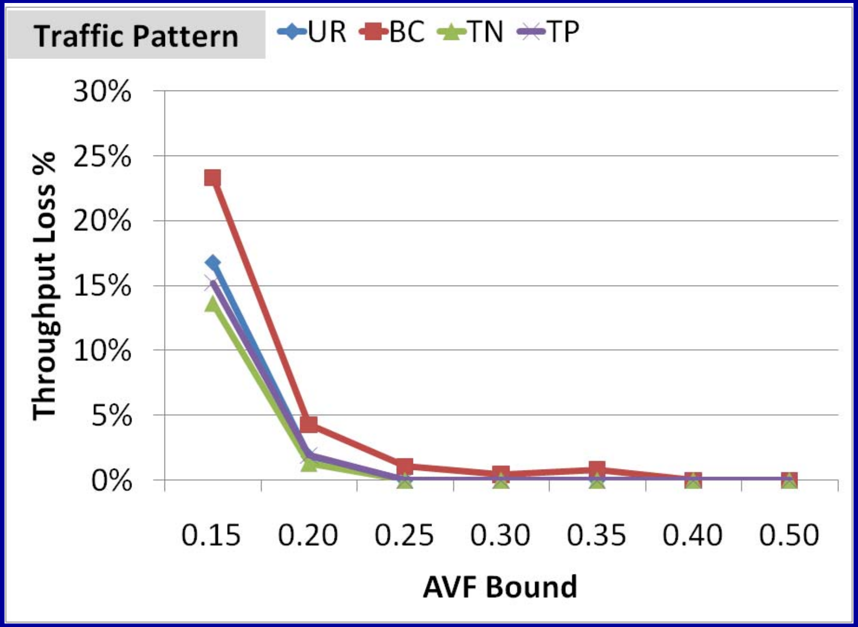
CYCLE 4

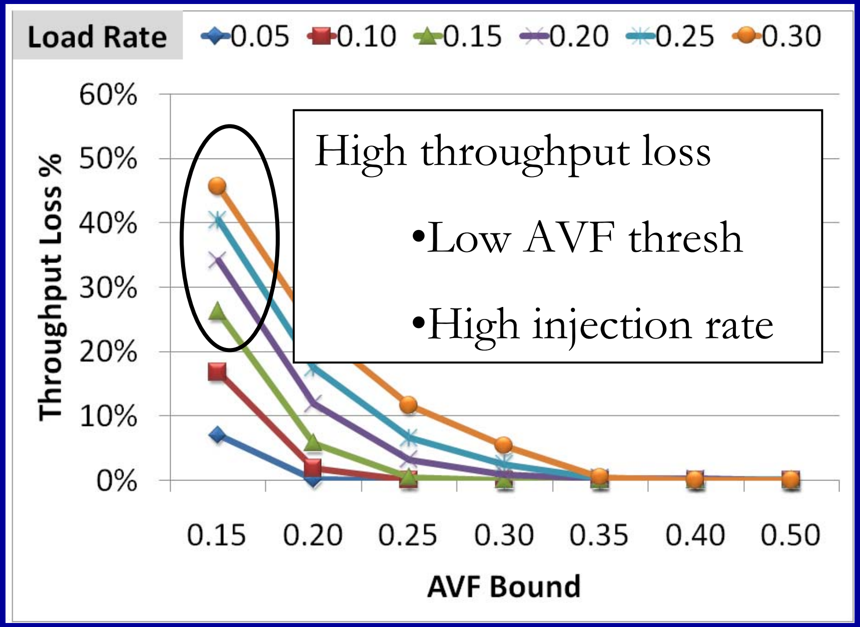
Power savings





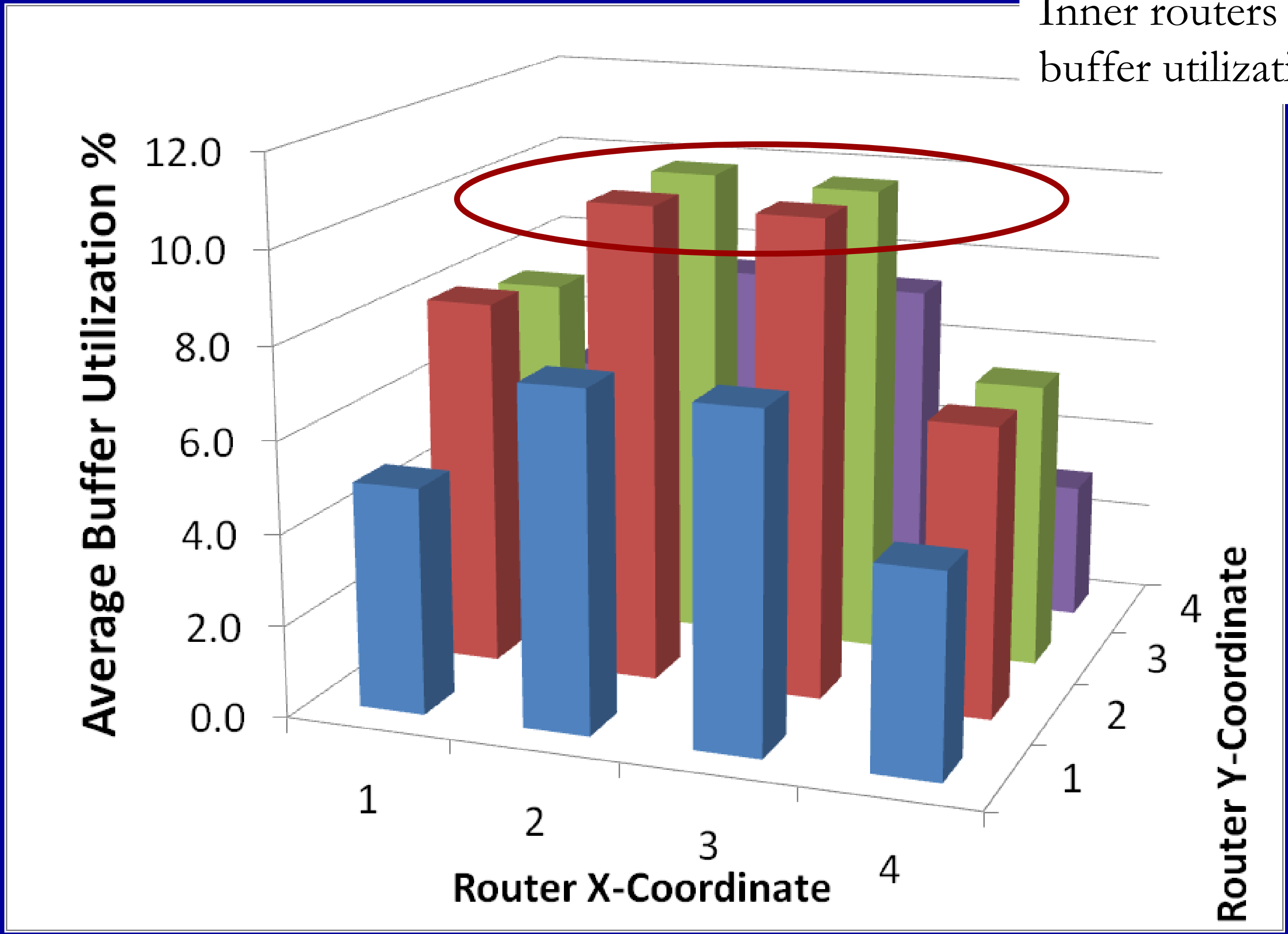
Throughput loss





Understanding throughput loss

Inner routers have higher buffer utilization.



Overview

- Motivation
- Contributions
- Analytical model
- **Implementation details and results**
  - SAVF results
  - **MAVF results**
- Conclusion

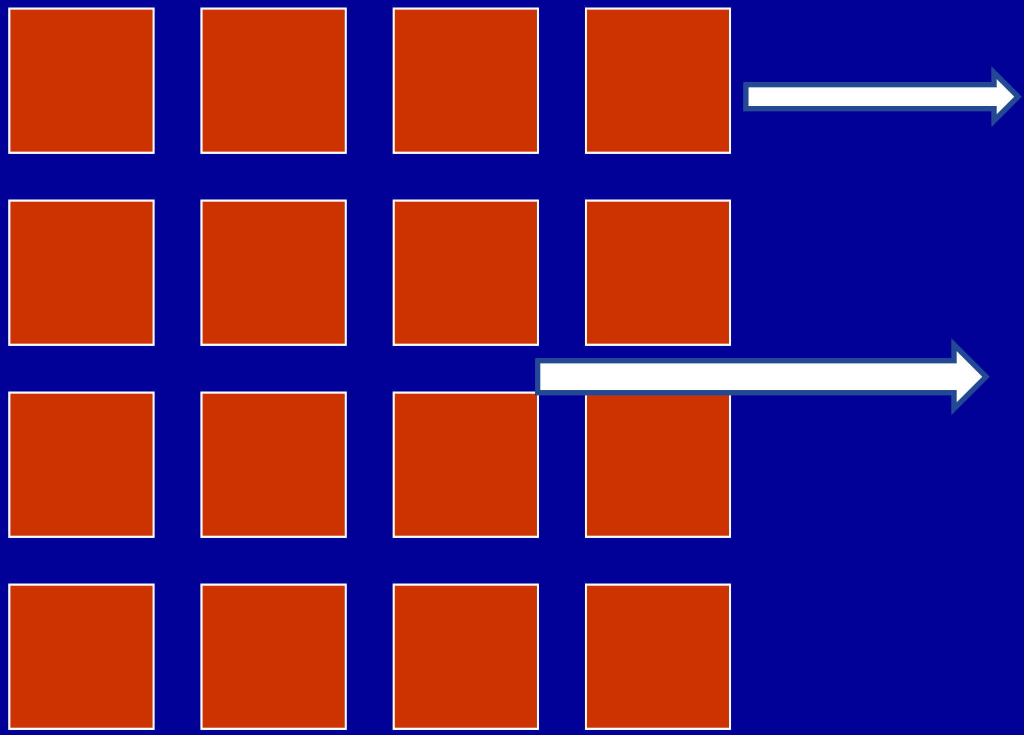
Multiple AVF domains (MAVF)

- Static MAVF
- Dynamic MAVF
- Both solutions are better than fixing the AVF of each router as SAVF scheme does

Static MAVF

- Static analysis of buffer utilization
- Same fixed overall AVF bound.
- Helps redistribute the AVF thresh to places where utilization is higher.
  - Outer routers have lesser threshold than the inner routers.
- **Ideal Static**. – Best possible. Allows all inter router movement but blocks new injections
  - Impractical to implement.

Static MAVF



Outer router

$\rho = 1/4$

Inner router

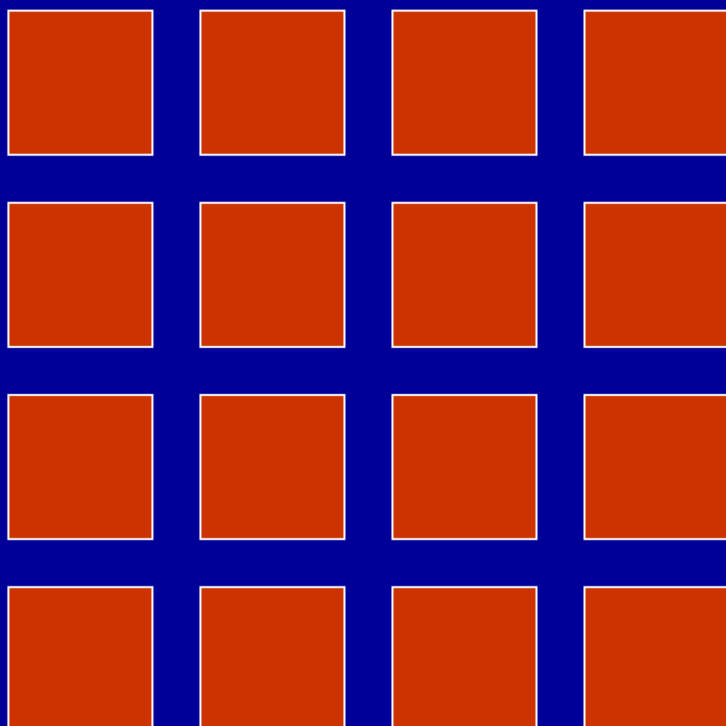
$\rho = 3/4$

The routers have different duty cycles
Preconfigured Statically

Dynamic MAVF

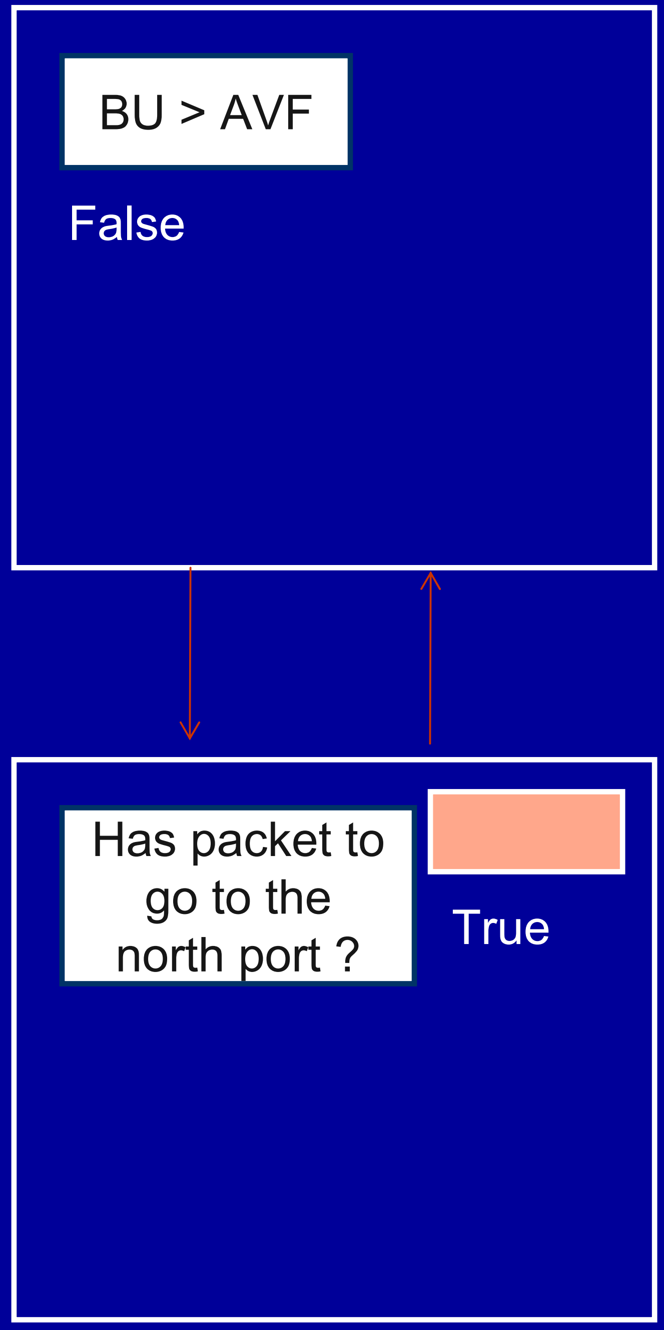
- Dynamically adapts to the changing requirements of the application
- Overall AVF_{thresh} can ***increase*** if the application needs it.
- No pressure to decide the optimal threshold through static analysis.
- AVF_{thresh} can change (increase) => Power savings can change (decrease).

Dynamic MAVF



Implementation : Case 1

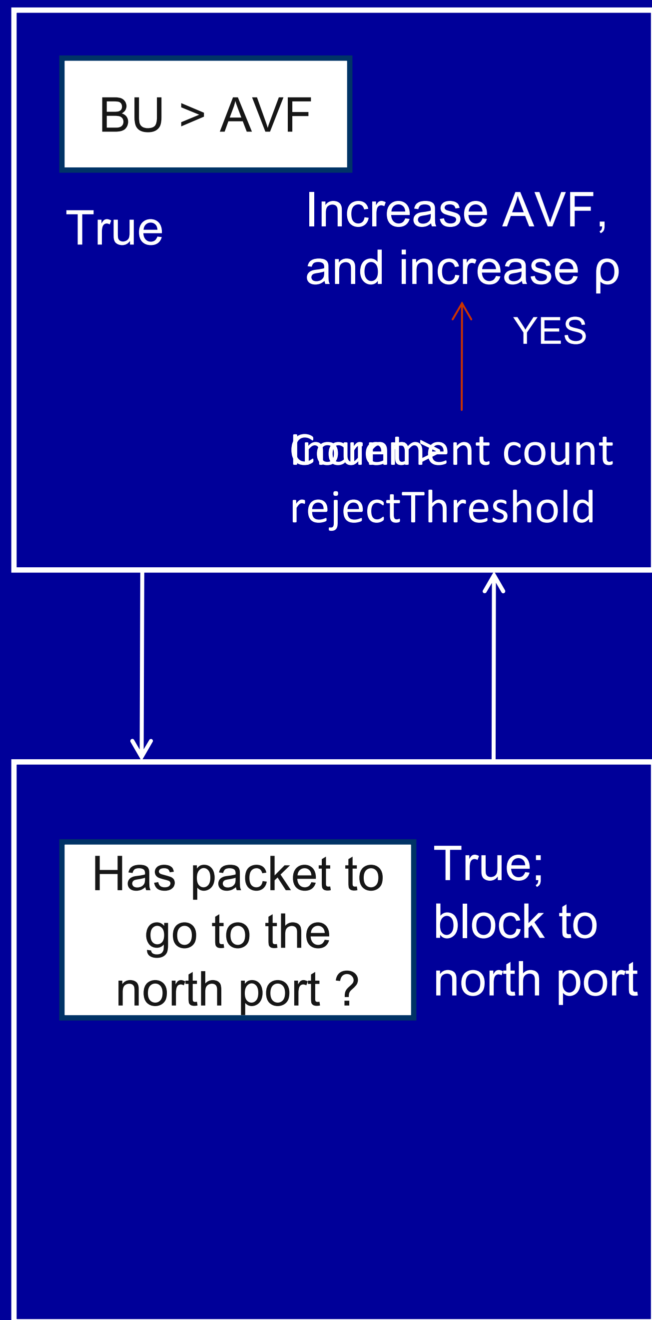
Cycle 0



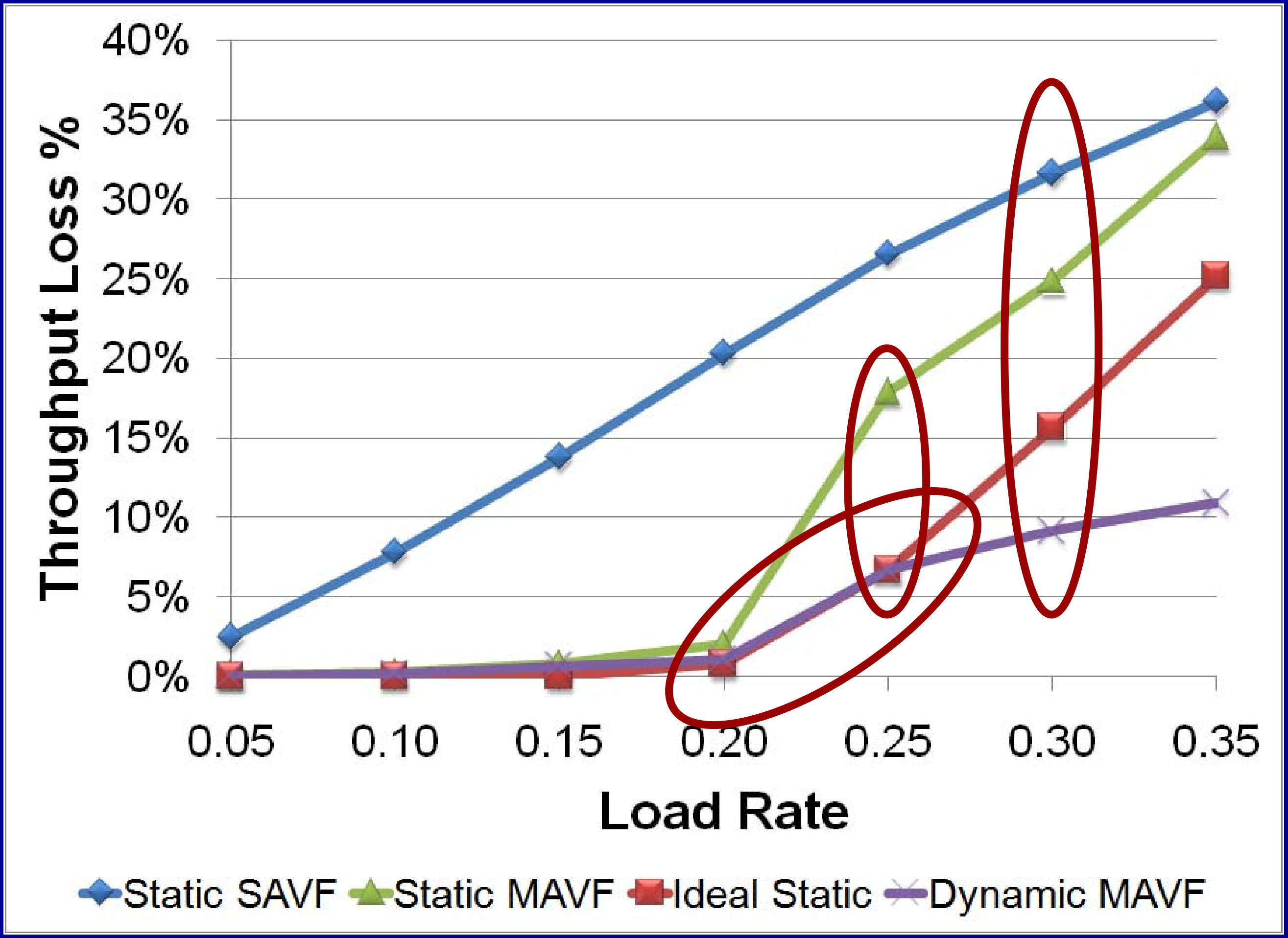
Implementation : Case 2

Cycle

0 1

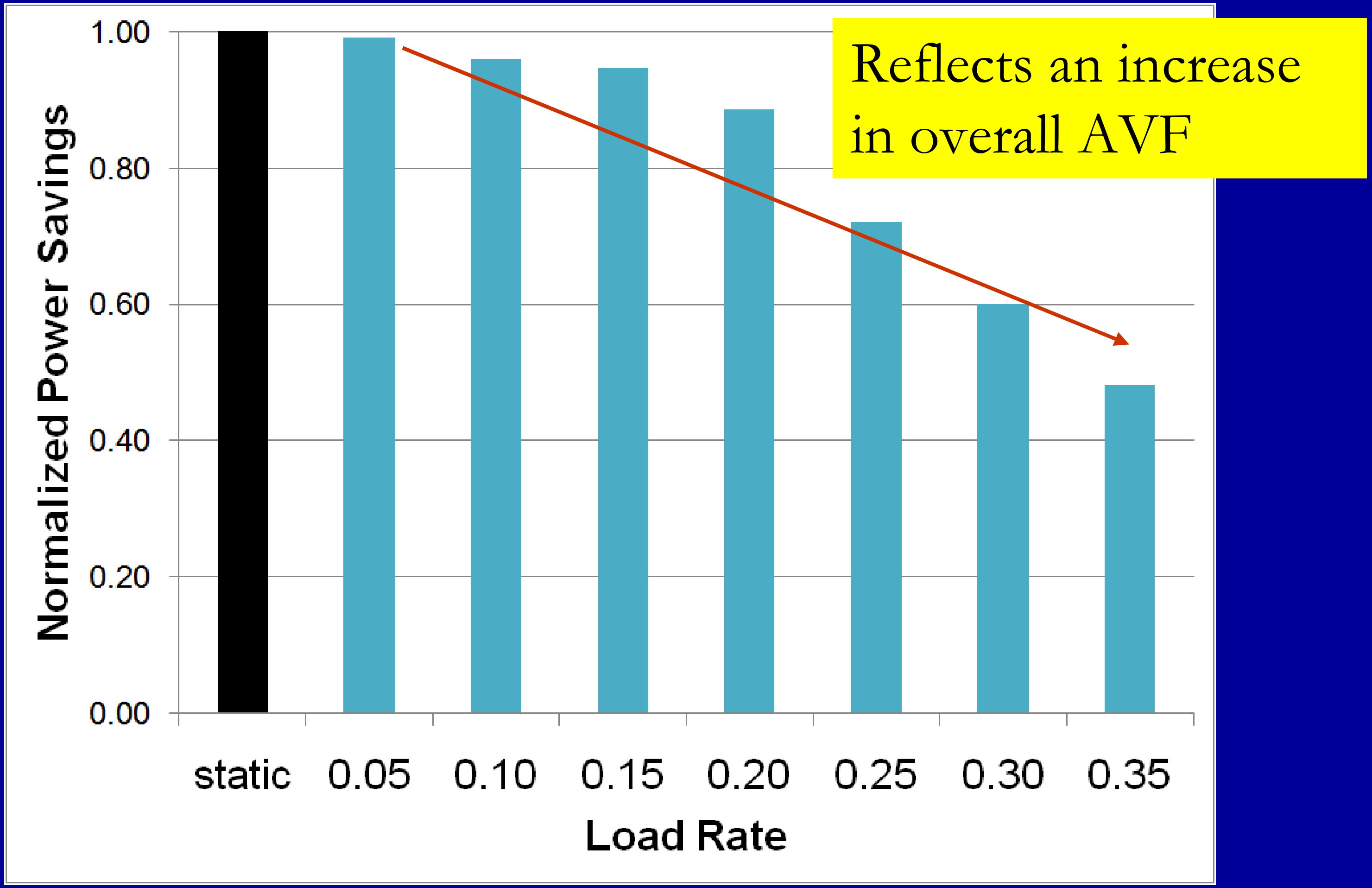


Throughput loss

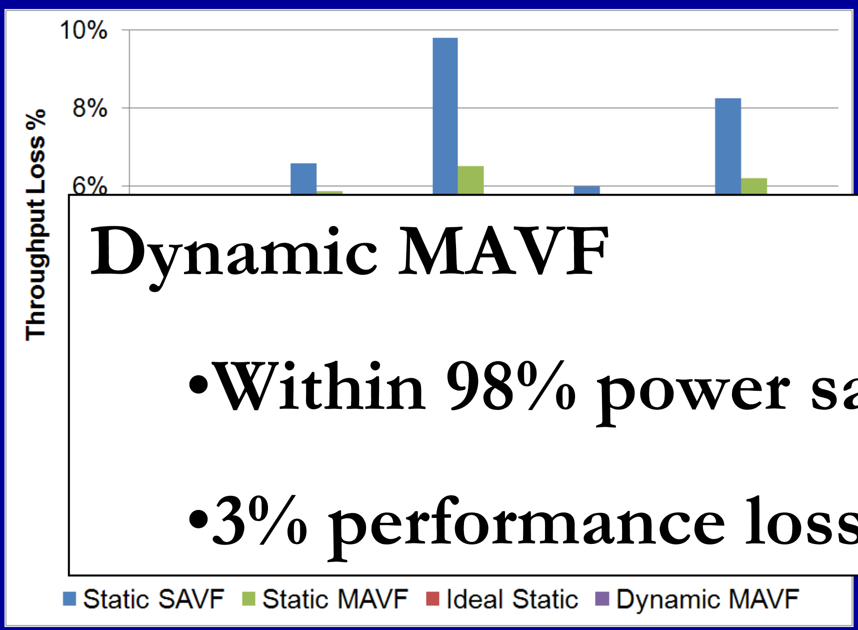


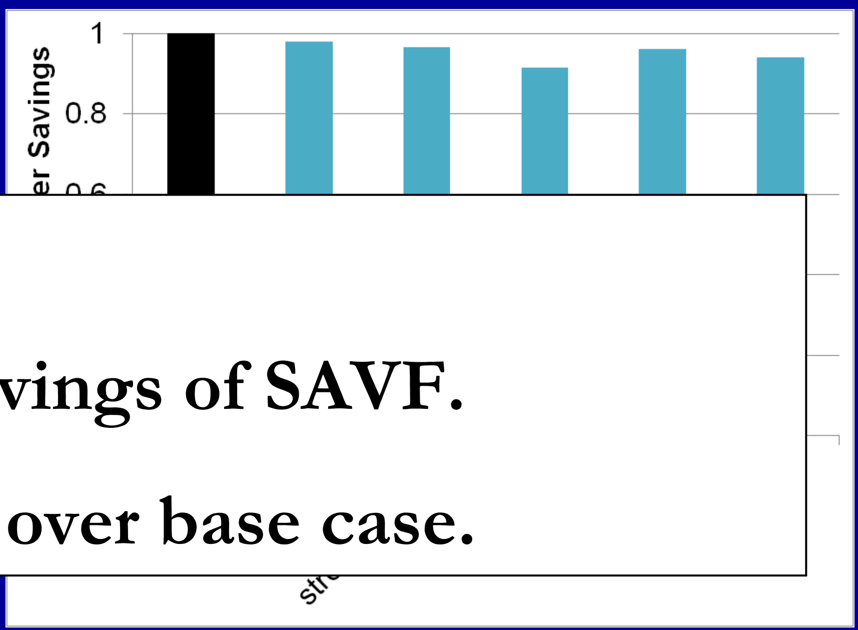
Dynamic MAVF performs well under high load rate

Dynamic MAVF Power savings



Application results





Dynamic MAVF

- Within 98% power savings of SAVF.
- 3% performance loss over base case.

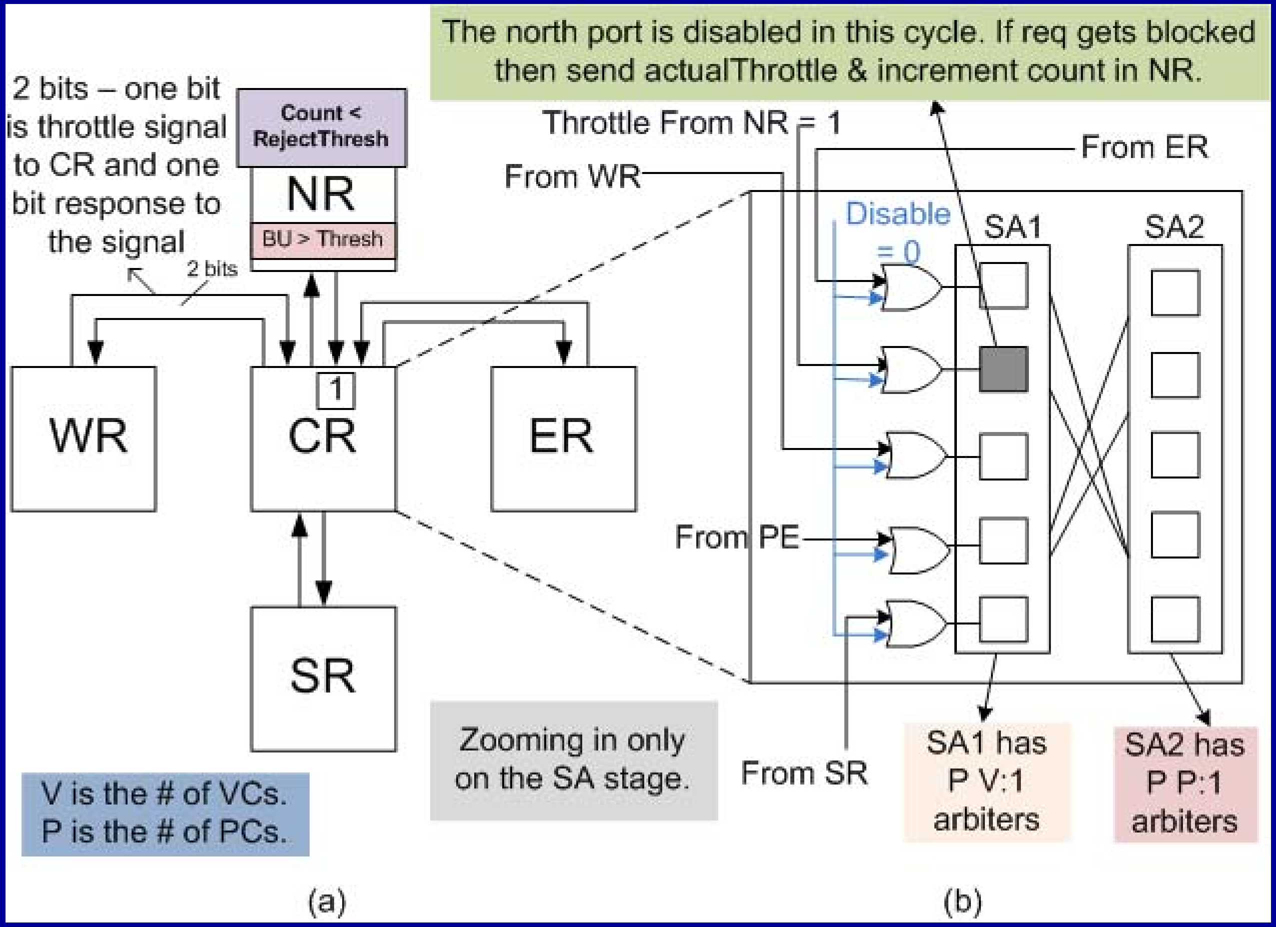
Conclusion

- Inherent fault tolerance of applications can lead to ECC power savings.
- AVF bounding increases power savings while incurring throughput loss.
- MAVF schemes decrease the throughput loss.
  - 44% of ECC power can be saved at a throughput loss of 3%
- In future, we would like to model for multi-bit errors.

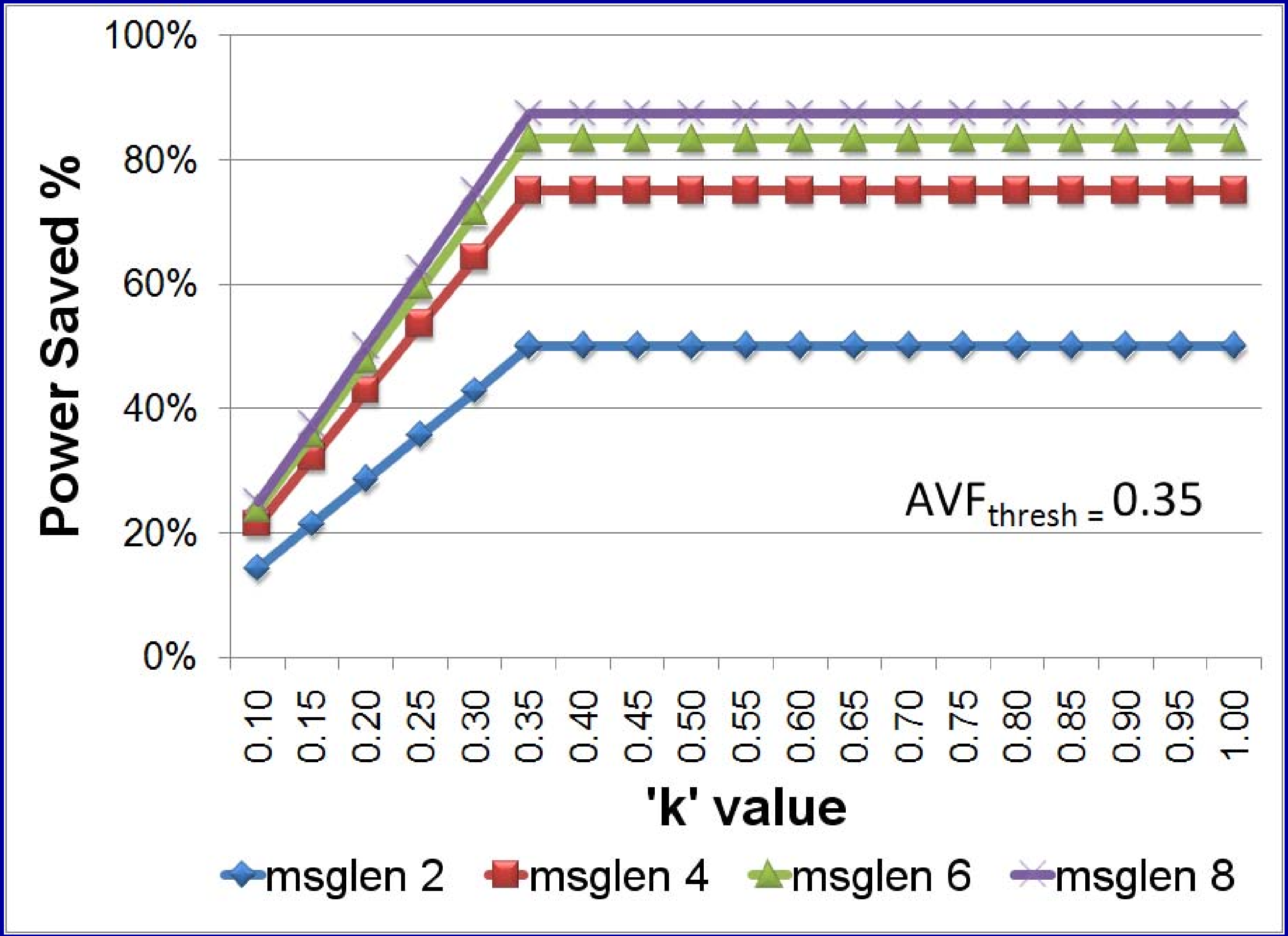
THAT YOU

BACKUP

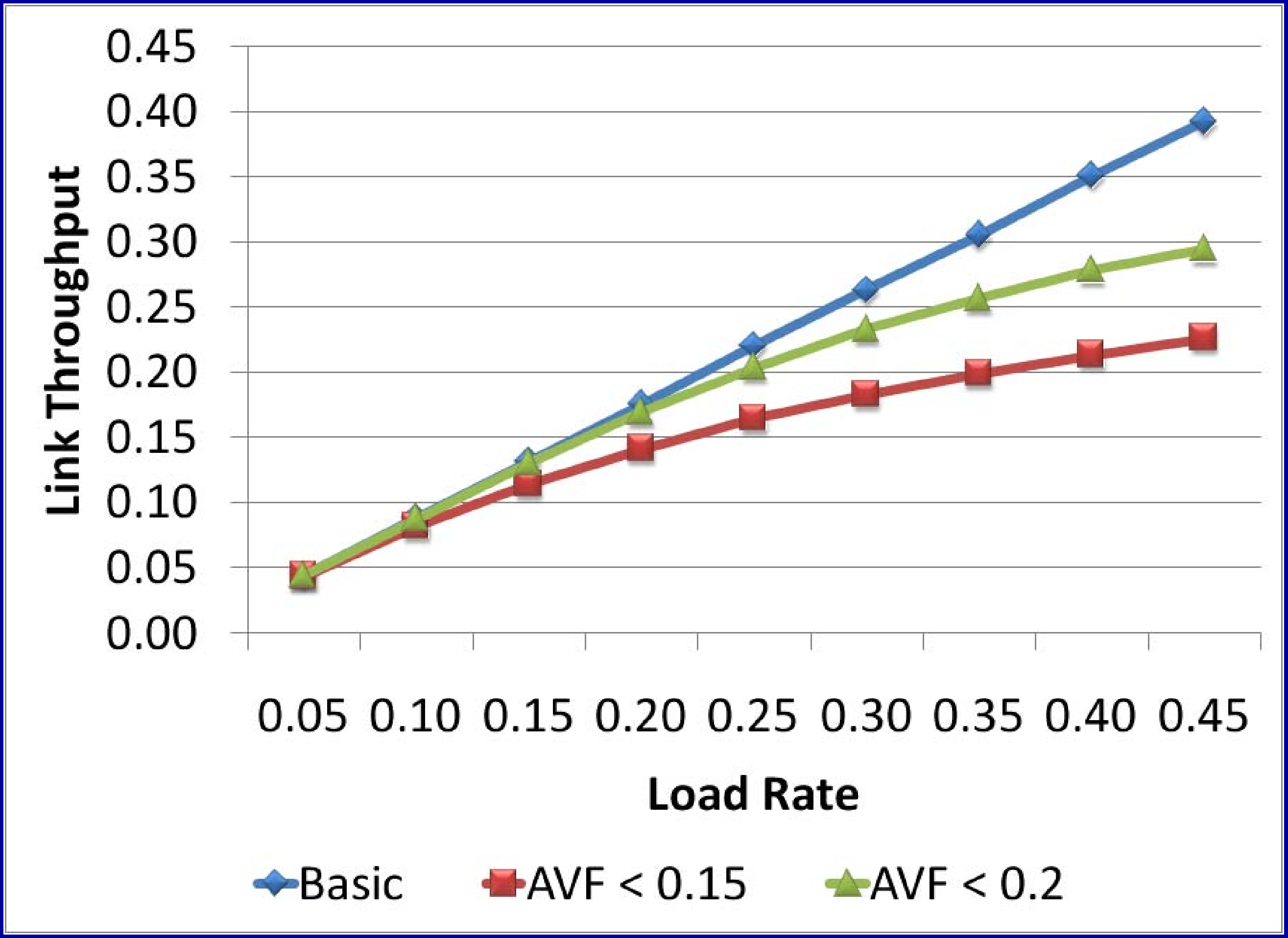
Architectural Details



SAVF power savings



SAVF link throughput



Net AVF is Dynamic MAVF