TCP/IP in hardware using SME

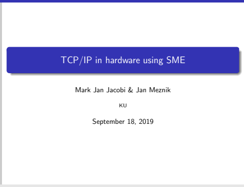
Mark Jan Jacobi & Jan Meznik

KU

September 18, 2019

2019-09-18

TCP/IP in hardware using SME



TCP/IP in hardware using SME

Mark Jan Jacobi & Jan Meznik

KU

September 18, 2019

Mark siger introduktion og 2-3 saetninger "abstrakt"

Table of Contents

2019-09-18

TCP/IP in hardware using SME
└Introduction
└Table of Contents

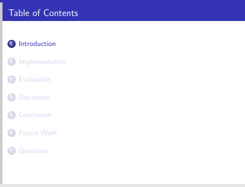
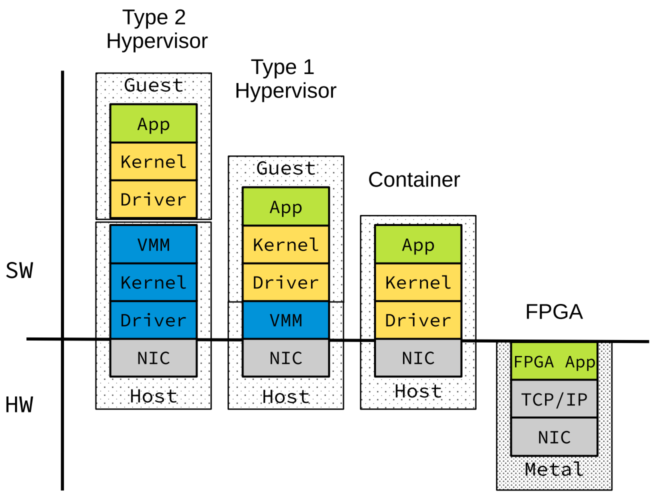
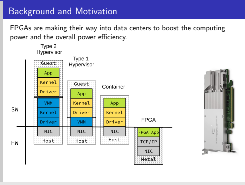


Table of Contents

Background and Motivation

FPGAs are making their way into data centers to boost the computing power and the overall power efficiency.





Applikationer og Big-Data udregninger flytter til Cloud, drevet af store data centre.

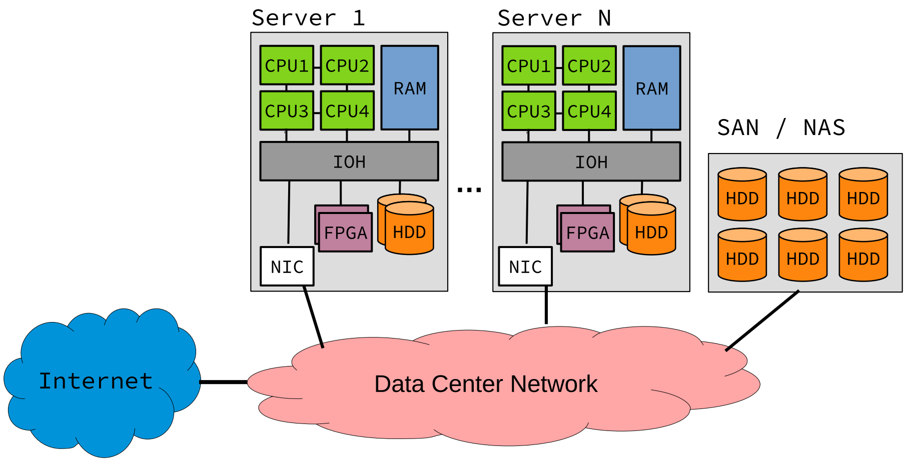
Disse data-centre kraever rigtigt meget plads, store maengder af stroem og er i stigende grad svaere at vedligeholde og udvide.

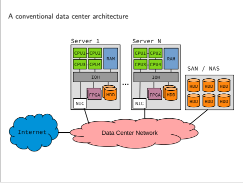
De fleste data-centre er derfor begyndt at aflaste beregningerne til FPGAer, som fjerner meget af overhead til beregningerne

kan bruges til at få en computer til at køre hurtigere hvis de mest brugte instruktioner, skrives direkte ned i hardwaren

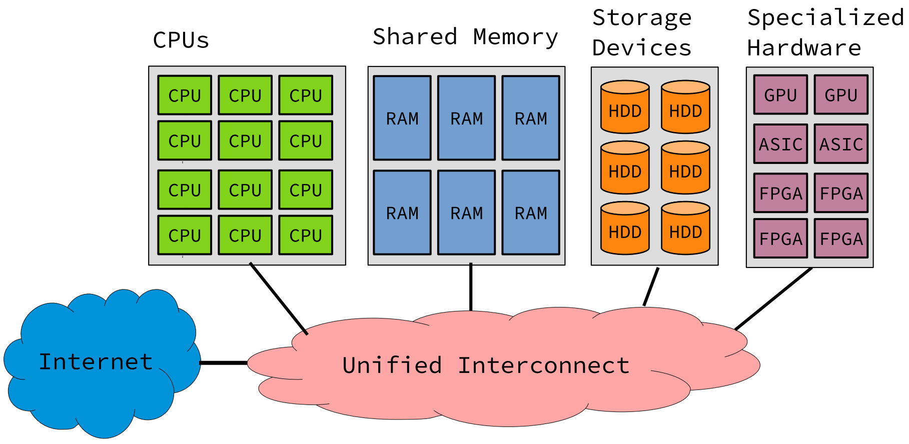
PROBLEMET er at der kun kan vaere en begreanset antal af FPGAer i konventionele servere

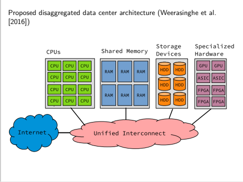
A conventional data center architecture





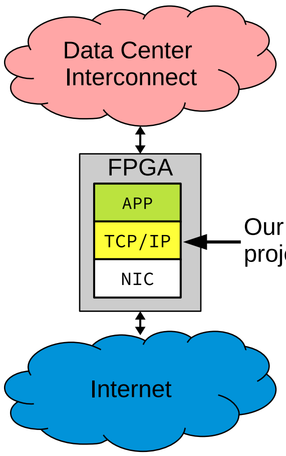
Proposed disaggregated data center architecture (Weerasinghe et al. [2016])



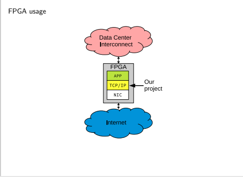


Hvis man splitter resourcerne op, kan man takket været FPGA få bedre ydeevne på det samme areal, samt nemmere håndtering af servere og deres komponenter.

FPGA usage



TCP/IP in hardware using SME
└─Introduction



The Internet

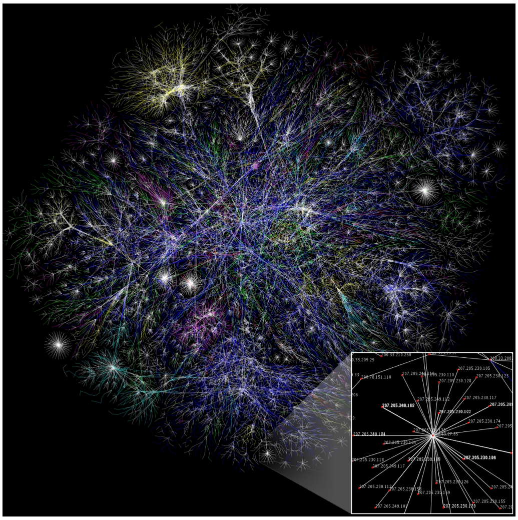
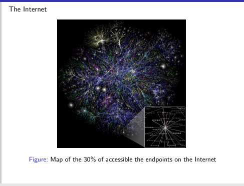


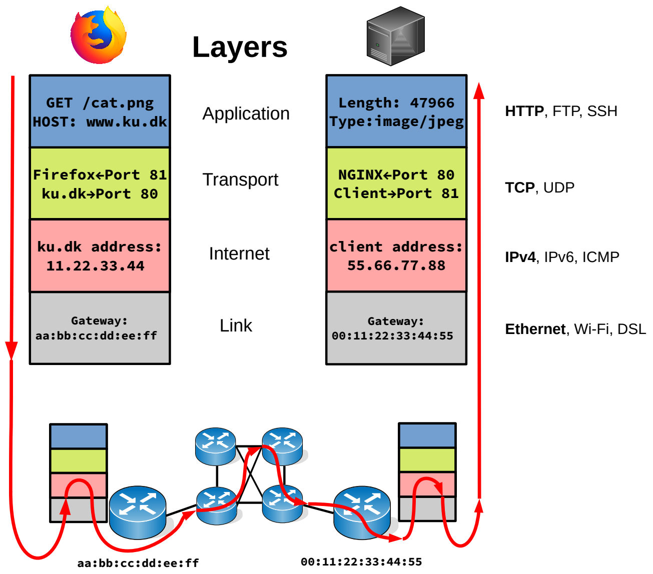
Figure: Map of the 30% of accessible the endpoints on the Internet



The Internet

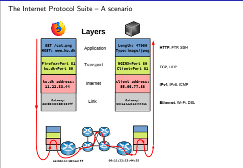
Figure: Map of the 30% of accessible the endpoints on the Internet

The Internet Protocol Suite – A scenario

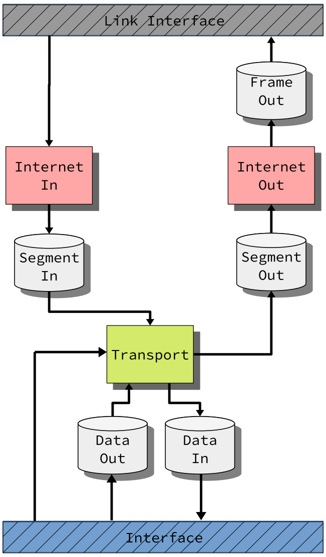


TCP/IP in hardware using SME
└Introduction

2019-09-18



Design with the 4 layers in mind



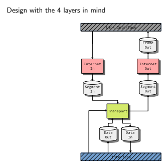


Table of Contents

2019-09-18

TCP/IP in hardware using SME
└─Implementation
  └─Table of Contents

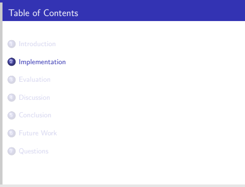
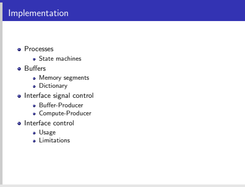


Table of Contents

1. Introduction
2. Implementation
3. Evaluation
4. Discussion
5. Conclusion
6. Future Work
7. Questions

Implementation

- Processes
  - State machines
- Buffers
  - Memory segments
  - Dictionary
- Interface signal control
  - Buffer-Producer
  - Compute-Producer
- Interface control
  - Usage
  - Limitations


Implementation

- Processes
  - State machines
- Buffers
  - Memory segments
  - Dictionary
- Interface signal control
  - Buffer-Producer
  - Compute-Producer
- Interface control
  - Usage
  - Limitations

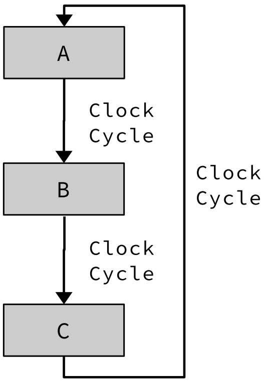
Implementation
Processes

State machines

```
public class SomeProcess :
↪   StateProcess
{
  private override async
  ↪   Task OnTickAsync()
  {
    a();
    await ClockAsync();
    b();
    await ClockAsync();
    c();
    await ClockAsync();
  }
}
```

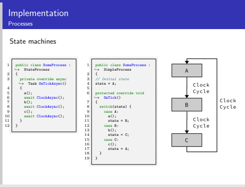
```
public class SomeProcess :
↪   SimpleProcess
{
// Initial state
state = A;

protected override void
↪   OnTick()
{
  switch(state) {
    case A:
      a();
      state = B;
    case B:
      b();
      state = C;
    case C:
      c();
      state = A;
  }
}
```



2019-09-18

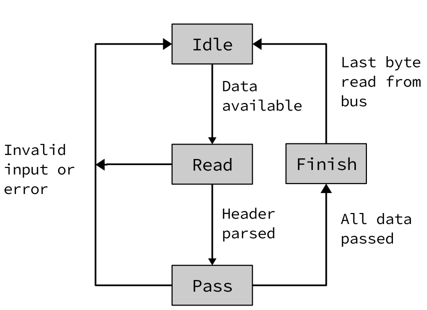
TCP/IP in hardware using SME
└Implementation
└Implementation

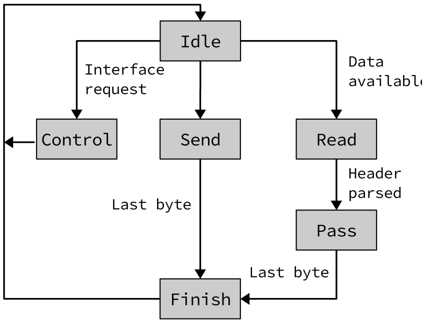

Implementation
Processes
State machines
public class SomeProcess : ↪ StateProcess { private override async ↪ Task OnTickAsync() { a(); await ClockAsync(); b(); await ClockAsync(); c(); await ClockAsync(); } }
public class SomeProcess : ↪ SimpleProcess { // Initial state state = A; protected override void ↪ OnTick() { switch(state) { case A: a(); state = B; case B: b(); state = C; case C: c(); state = A; } }
A Clock Cycle B Clock Cycle C Clock Cycle

Implementation

Processes

Examples

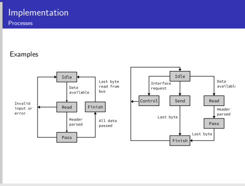




2019-09-18

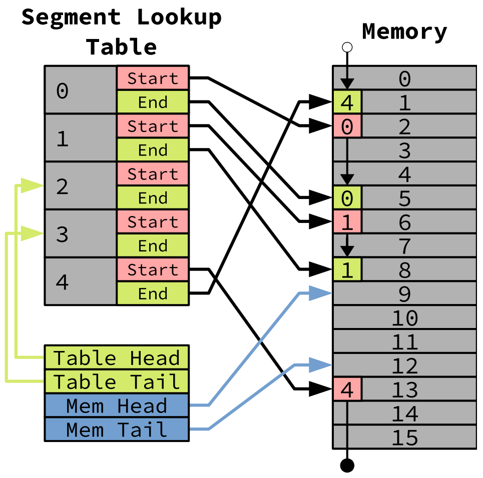
TCP/IP in hardware using SME
└Implementation

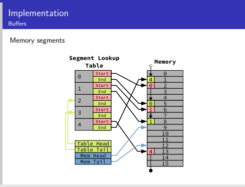
└Implementation



Implementation
Buffers

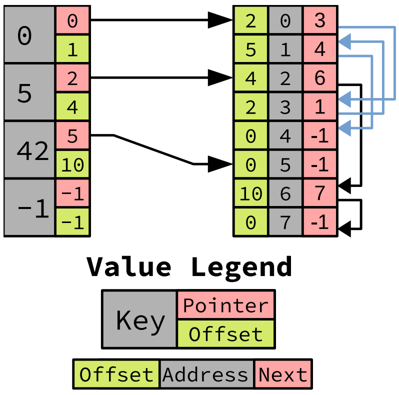
Memory segments





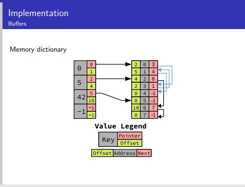
Implementation
Buffers

Memory dictionary



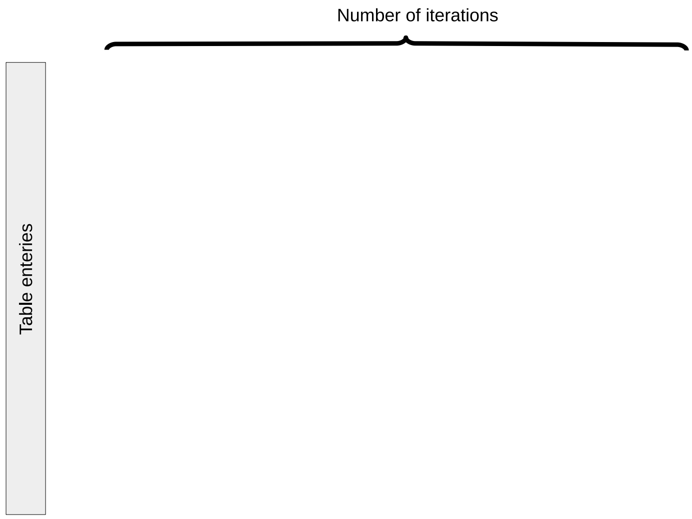
2019-09-18

TCP/IP in hardware using SME
└Implementation
└Implementation



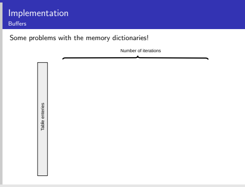
Implementation
Buffers

Some problems with the memory dictionaries!



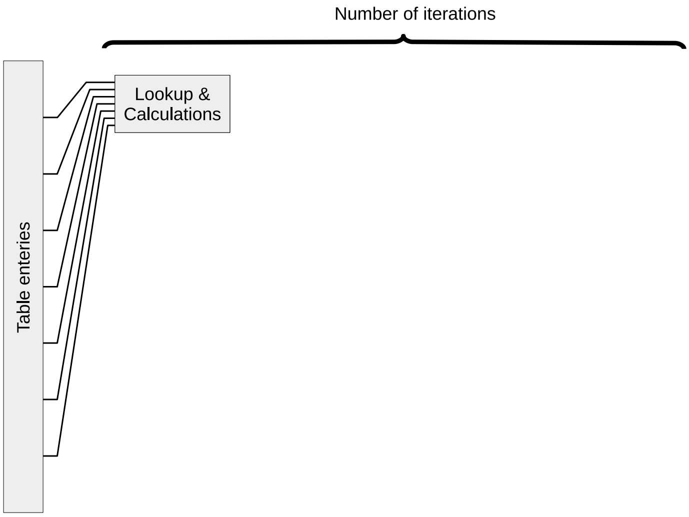
2019-09-18

TCP/IP in hardware using SME
└Implementation
└Implementation


Implementation
Buffers
Some problems with the memory dictionaries!

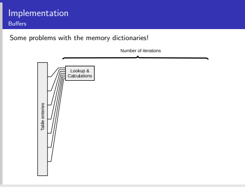
Implementation
Buffers

Some problems with the memory dictionaries!



2019-09-18

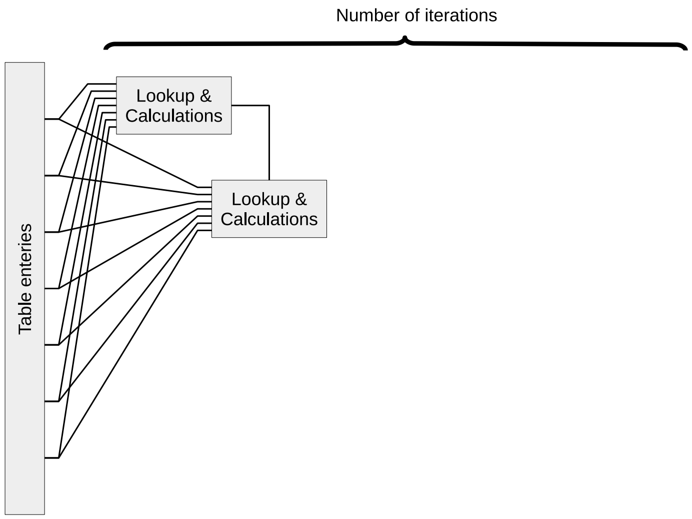
TCP/IP in hardware using SME
└Implementation
└Implementation



Implementation

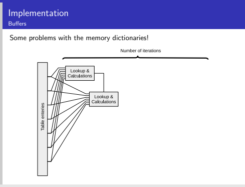
Buffers

Some problems with the memory dictionaries!



2019-09-18

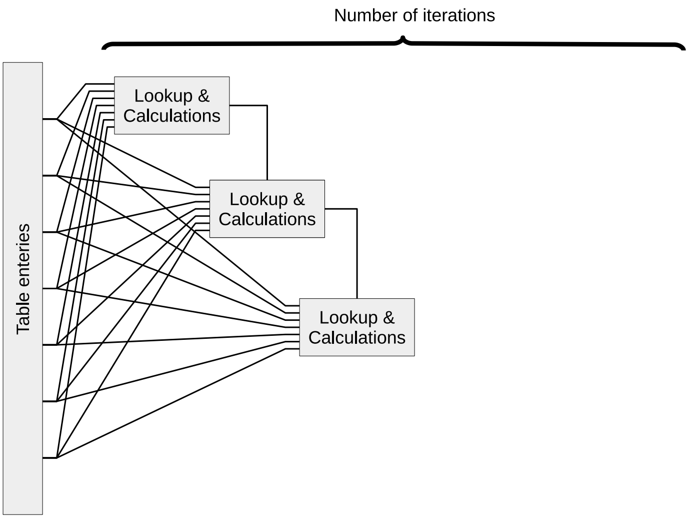
TCP/IP in hardware using SME
└Implementation
└Implementation


Implementation
Buffers
Some problems with the memory dictionaries!

Implementation

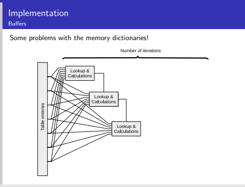
Buffers

Some problems with the memory dictionaries!



2019-09-18

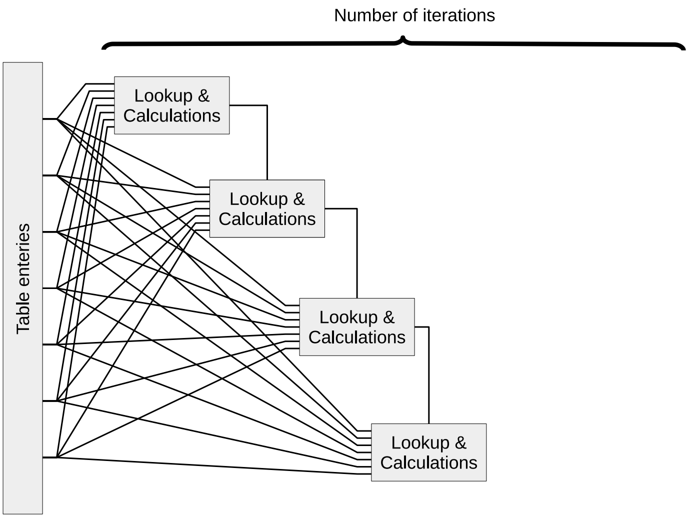
TCP/IP in hardware using SME
└Implementation
└Implementation



Implementation

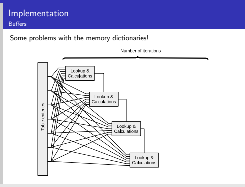
Buffers

Some problems with the memory dictionaries!



2019-09-18

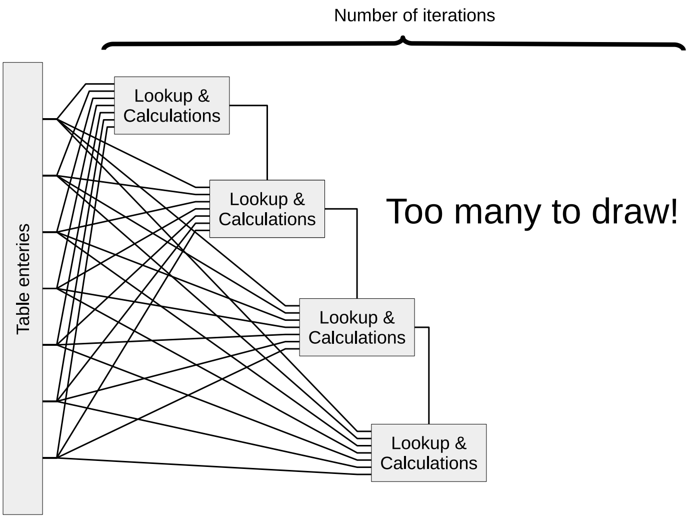
TCP/IP in hardware using SME
└Implementation
└Implementation



Implementation

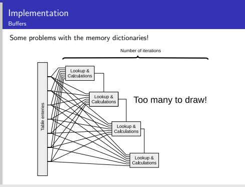
Buffers

Some problems with the memory dictionaries!



2019-09-18

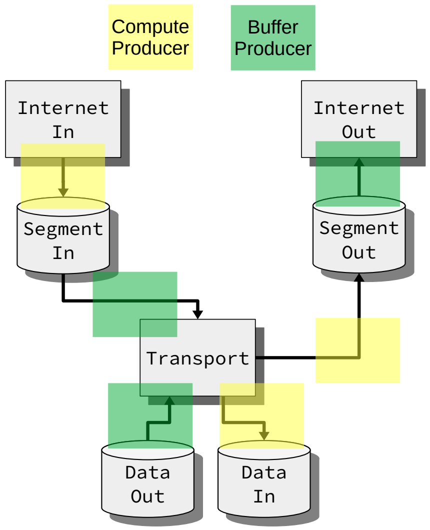
TCP/IP in hardware using SME
└─Implementation
└─Implementation



Implementation

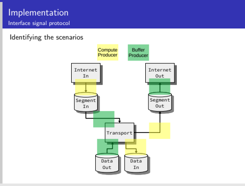
Interface signal protocol

Identifying the scenarios



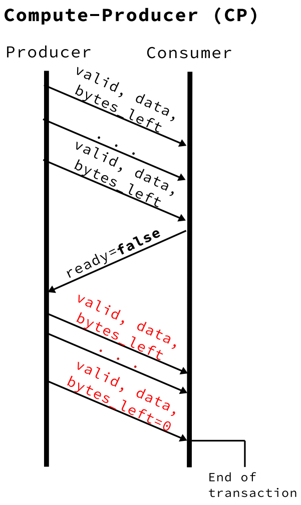
2019-09-18

TCP/IP in hardware using SME
└─Implementation
└─Implementation



Implementation

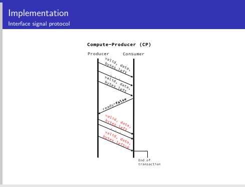
Interface signal protocol



2019-09-18

TCP/IP in hardware using SME
└Implementation

└Implementation

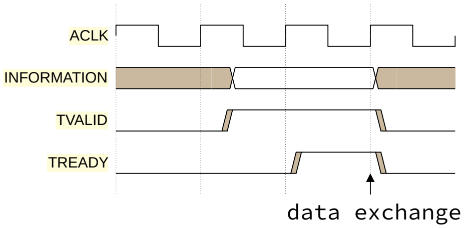


Implementation

Interface signal protocol

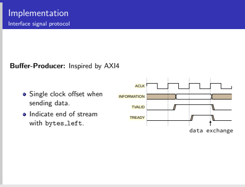
Buffer-Producer: Inspired by AXI4

- Single clock offset when sending data.
- Indicate end of stream with `bytes_left`.



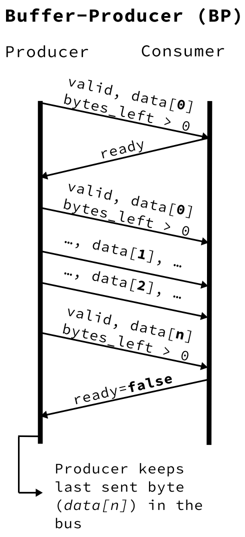
2019-09-18

TCP/IP in hardware using SME
└Implementation
    └Implementation



Implementation

Interface signal protocol



2019-09-18

TCP/IP in hardware using SME
└Implementation
└Implementation

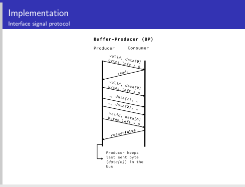


Table of Contents

2019-09-18

TCP/IP in hardware using SME
└Evaluation
└Table of Contents

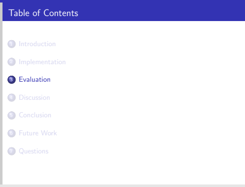
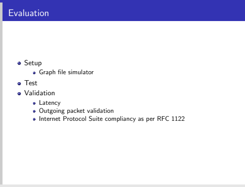


Table of Contents

Evaluation

- Setup
  - Graph file simulator
- Test
- Validation
  - Latency
  - Outgoing packet validation
  - Internet Protocol Suite compliancy as per RFC 1122

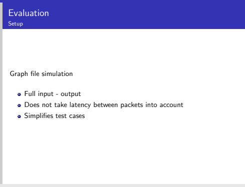

Evaluation

- Setup
  - Graph file simulator
- Test
- Validation
  - Latency
  - Outgoing packet validation
  - Internet Protocol Suite compliancy as per RFC 1122

Evaluation
Setup

Graph file simulation

- Full input - output
- Does not take latency between packets into account
- Simplifies test cases


Evaluation
Setup

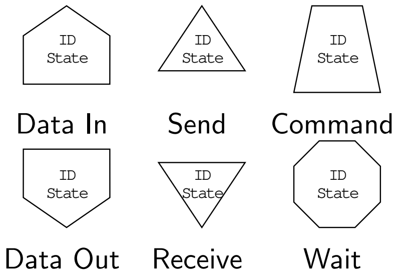
Graph file simulation

- Full input - output
- Does not take latency between packets into account
- Simplifies test cases

Evaluation

Setup

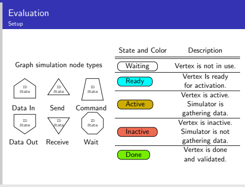
Graph simulation node types



State and Color	Description
Waiting	Vertex is not in use.
Ready	Vertex Is ready for activation.
Active	Vertex is active. Simulator is gathering data.
Inactive	Vertex is inactive. Simulator is not gathering data.
Done	Vertex is done and validated.

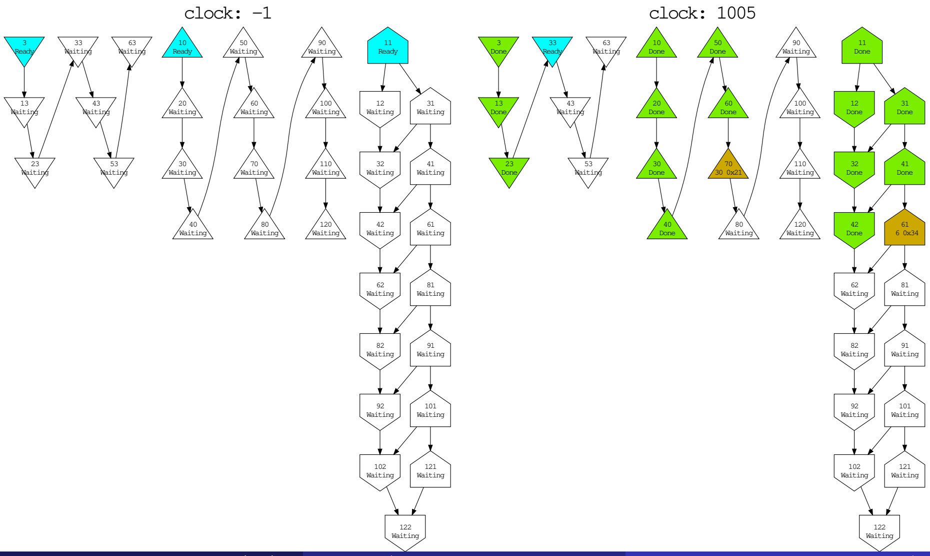
2019-09-18

TCP/IP in hardware using SME
└Evaluation
└Evaluation



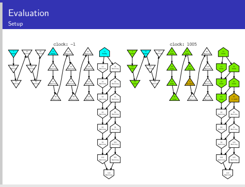
Evaluation

Setup



2019-09-18

TCP/IP in hardware using SME
└─Evaluation
└─Evaluation



Evaluation

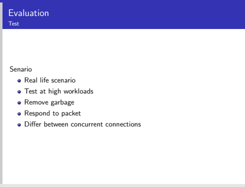
Test

Senario

- Real life scenario
- Test at high workloads
- Remove garbage
- Respond to packet
- Differ between concurrent connections

2019-09-18

TCP/IP in hardware using SME
└Evaluation
└Evaluation

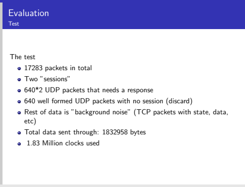

Evaluation
Test

Senario
- Real life scenario
- Test at high workloads
- Remove garbage
- Respond to packet
- Differ between concurrent connections

Evaluation
Test

The test

- 17283 packets in total
- Two "sessions"
- 640*2 UDP packets that needs a response
- 640 well formed UDP packets with no session (discard)
- Rest of data is "background noise" (TCP packets with state, data, etc)
- Total data sent through: 1832958 bytes
- 1.83 Million clocks used


Evaluation
Test
The test
- 17283 packets in total
- Two "sessions"
- 640*2 UDP packets that needs a response
- 640 well formed UDP packets with no session (discard)
- Rest of data is "background noise" (TCP packets with state, data, etc)
- Total data sent through: 1832958 bytes
- 1.83 Million clocks used

Evaluation
Validation

Latency calculations:

$n_\mathtt{D}$	:	The number of bytes in the data part of the protocol. This excludes both headers from transport and internet.
$n_\mathtt{I}$	:	The internet header size.
$n_\mathtt{T}$	:	The transport header size.
n	:	The total packet size.

From packet to user

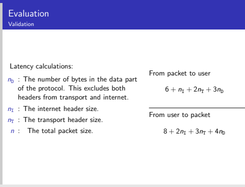
$$6 + n_\mathtt{I} + 2n_\mathtt{T} + 3n_\mathtt{D}$$

From user to packet

$$8 + 2n_\mathtt{I} + 3n_\mathtt{T} + 4n_\mathtt{D}$$

2019-09-18

TCP/IP in hardware using SME
└Evaluation
└Evaluation


Evaluation
Validation

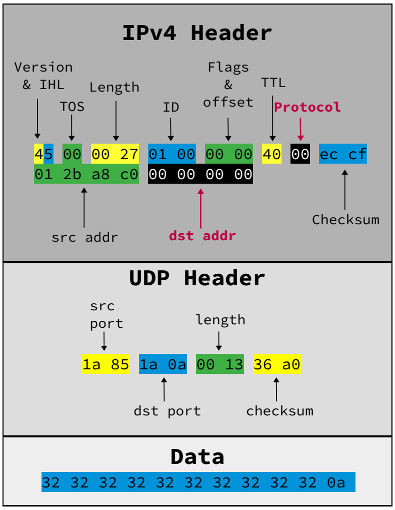
Latency calculations:
$n_\mathtt{D}$: The number of bytes in the data part of the protocol. This excludes both headers from transport and internet.
$n_\mathtt{I}$: The internet header size.
$n_\mathtt{T}$: The transport header size.
n : The total packet size.

From packet to user
$6 + n_\mathtt{I} + 2n_\mathtt{T} + 3n_\mathtt{D}$

From user to packet
$8 + 2n_\mathtt{I} + 3n_\mathtt{T} + 4n_\mathtt{D}$

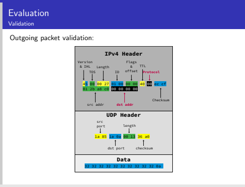
Evaluation
Validation

Outgoing packet validation:



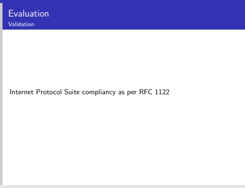
2019-09-18

TCP/IP in hardware using SME
└Evaluation
└Evaluation



Evaluation
Validation

Internet Protocol Suite compliancy as per RFC 1122


Evaluation
Validation

Internet Protocol Suite compliancy as per RFC 1122

Table of Contents

2019-09-18

TCP/IP in hardware using SME
└Discussion
└Table of Contents

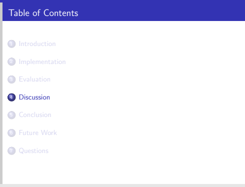
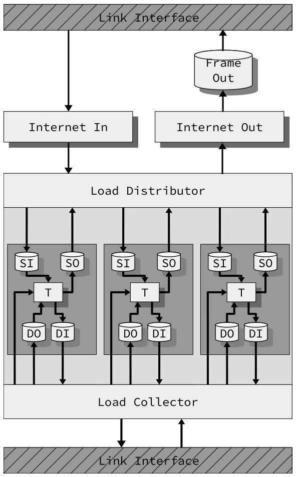


Table of Contents

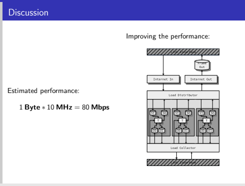
Discussion

Improving the performance:



Estimated performance:

$$1 \textbf{ Byte} * 10 \textbf{ MHz} = 80 \textbf{ Mbps}$$



Discussion

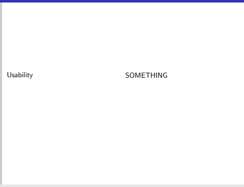
Improving the performance:

Estimated performance:

1 Byte * 10 MHz = 80 Mbps

Usability

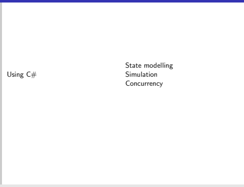
SOMETHING



Usability SOMETHING

Using C#

State modelling
Simulation
Concurrency



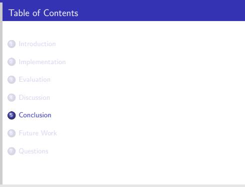
Using C#

State modelling
Simulation
Concurrency

Table of Contents

2019-09-18

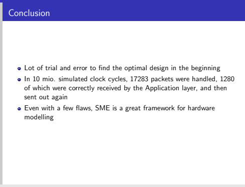
TCP/IP in hardware using SME
└─Conclusion
└─Table of Contents



Conclusion

- Lot of trial and error to find the optimal design in the beginning
- In 10 mio. simulated clock cycles, 17283 packets were handled, 1280 of which were correctly received by the Application layer, and then sent out again
- Even with a few flaws, SME is a great framework for hardware modelling

└Conclusion


Conclusion

- Lot of trial and error to find the optimal design in the beginning
- In 10 mio. simulated clock cycles, 17283 packets were handled, 1280 of which were correctly received by the Application layer, and then sent out again
- Even with a few flaws, SME is a great framework for hardware modelling

Table of Contents

2019-09-18

TCP/IP in hardware using SME
└Future Work
└Table of Contents

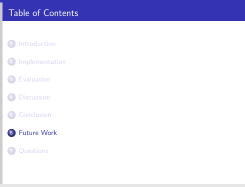
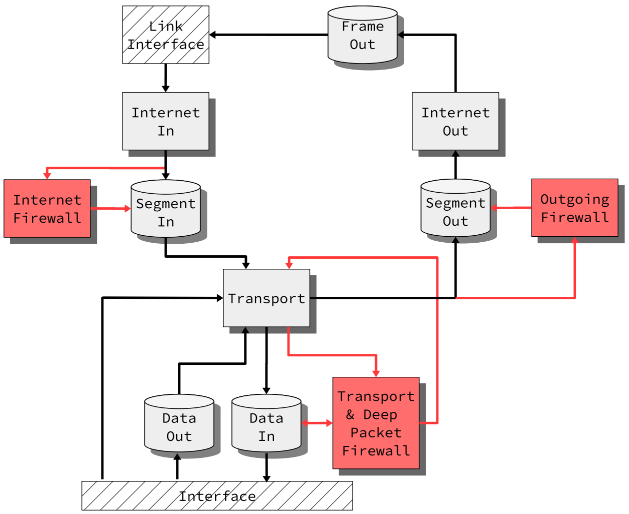


Table of Contents

1 Introduction
2 Implementation
3 Evaluation
4 Discussion
5 Conclusion
6 Future Work
7 Questions

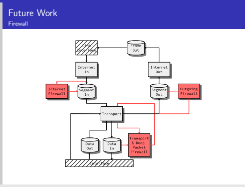
Future Work

Firewall



2019-09-18

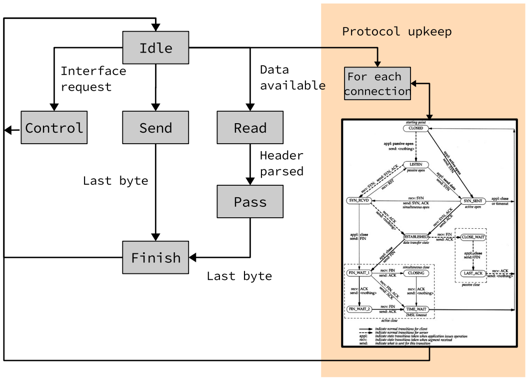
TCP/IP in hardware using SME
└Future Work
└Future Work



Future Work

TCP

Implementing TCP



2019-09-18

TCP/IP in hardware using SME
└Future Work
  └Future Work

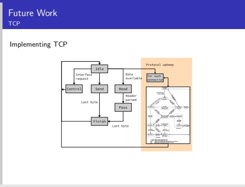


Table of Contents

2019-09-18

TCP/IP in hardware using SME
└Questions
└Table of Contents

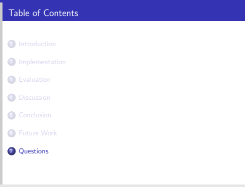
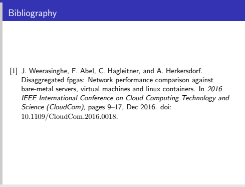


Table of Contents

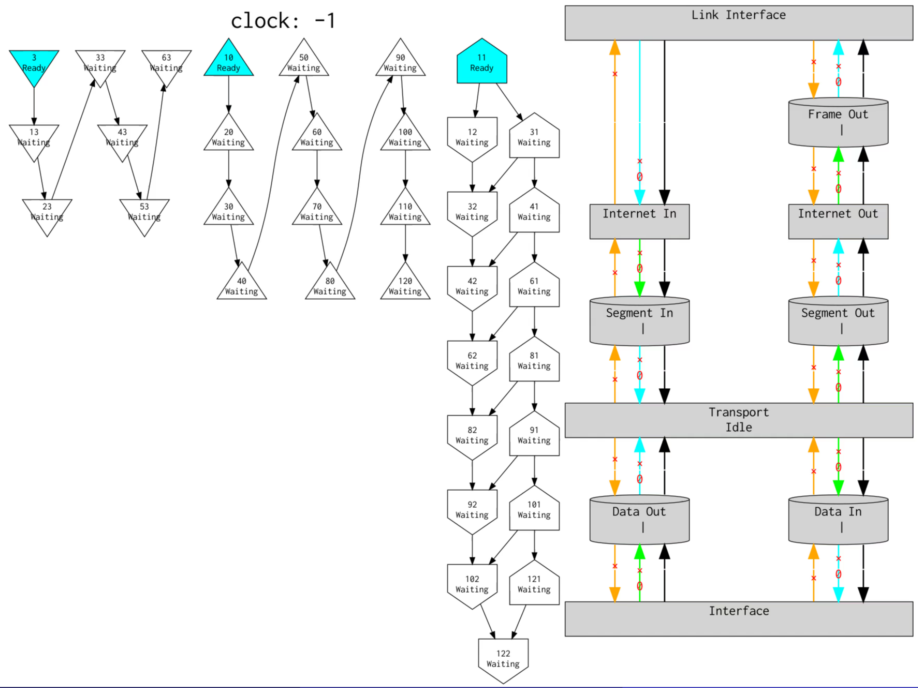
1 Introduction
2 Implementation
3 Evaluation
4 Discussion
5 Conclusion
6 Future Work
7 Questions

Bibliography

[1] J. Weerasinghe, F. Abel, C. Hagleitner, and A. Herkersdorf. Disaggregated fpgas: Network performance comparison against bare-metal servers, virtual machines and linux containers. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 9–17, Dec 2016. doi: 10.1109/CloudCom.2016.0018.


Bibliography

[1] J. Weerasinghe, F. Abel, C. Hagleitner, and A. Herkersdorf. Disaggregated fpgas: Network performance comparison against bare-metal servers, virtual machines and linux containers. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 9–17, Dec 2016. doi: 10.1109/CloudCom.2016.0018.



TCP/IP in hardware using SME
└Questions