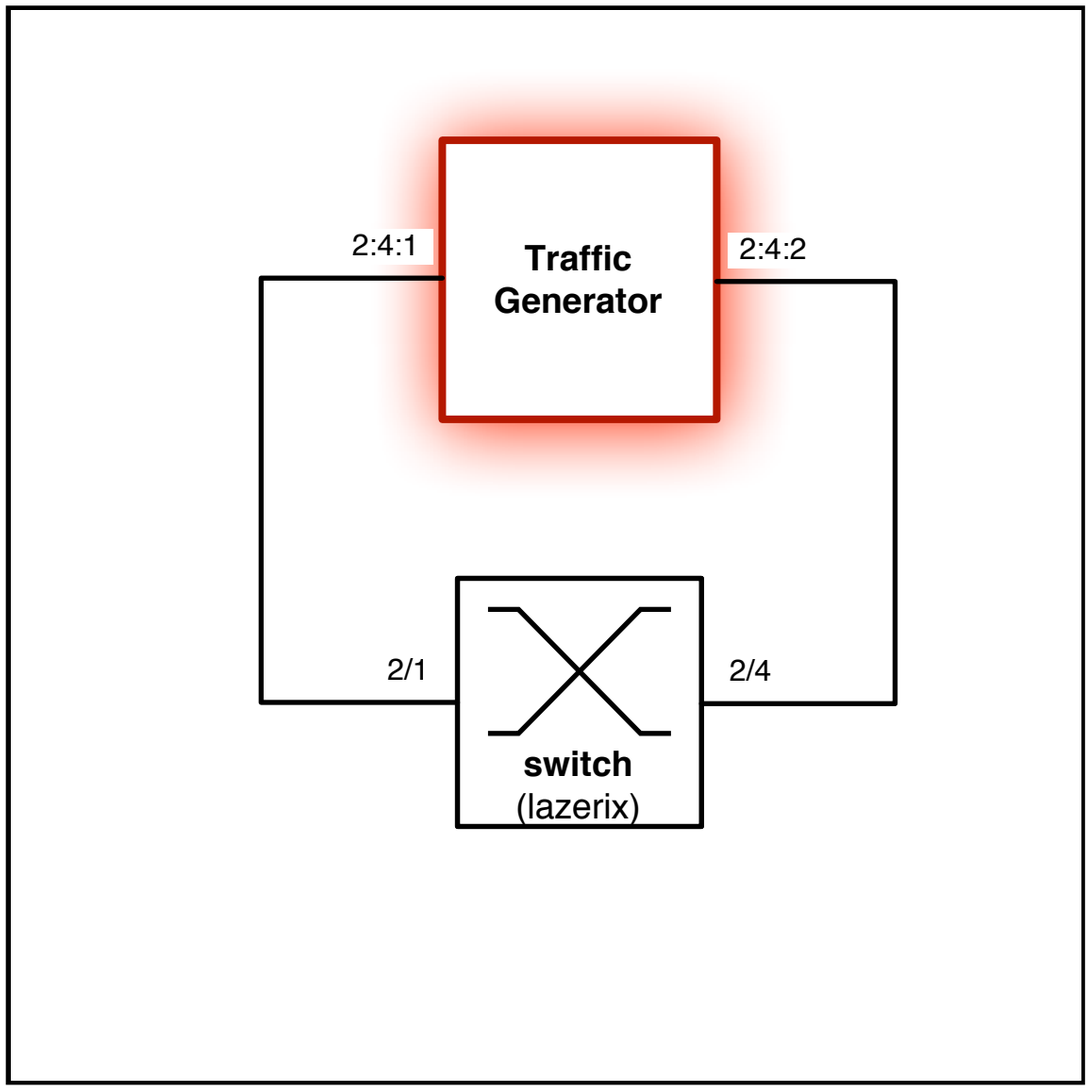
10GbE monitoring

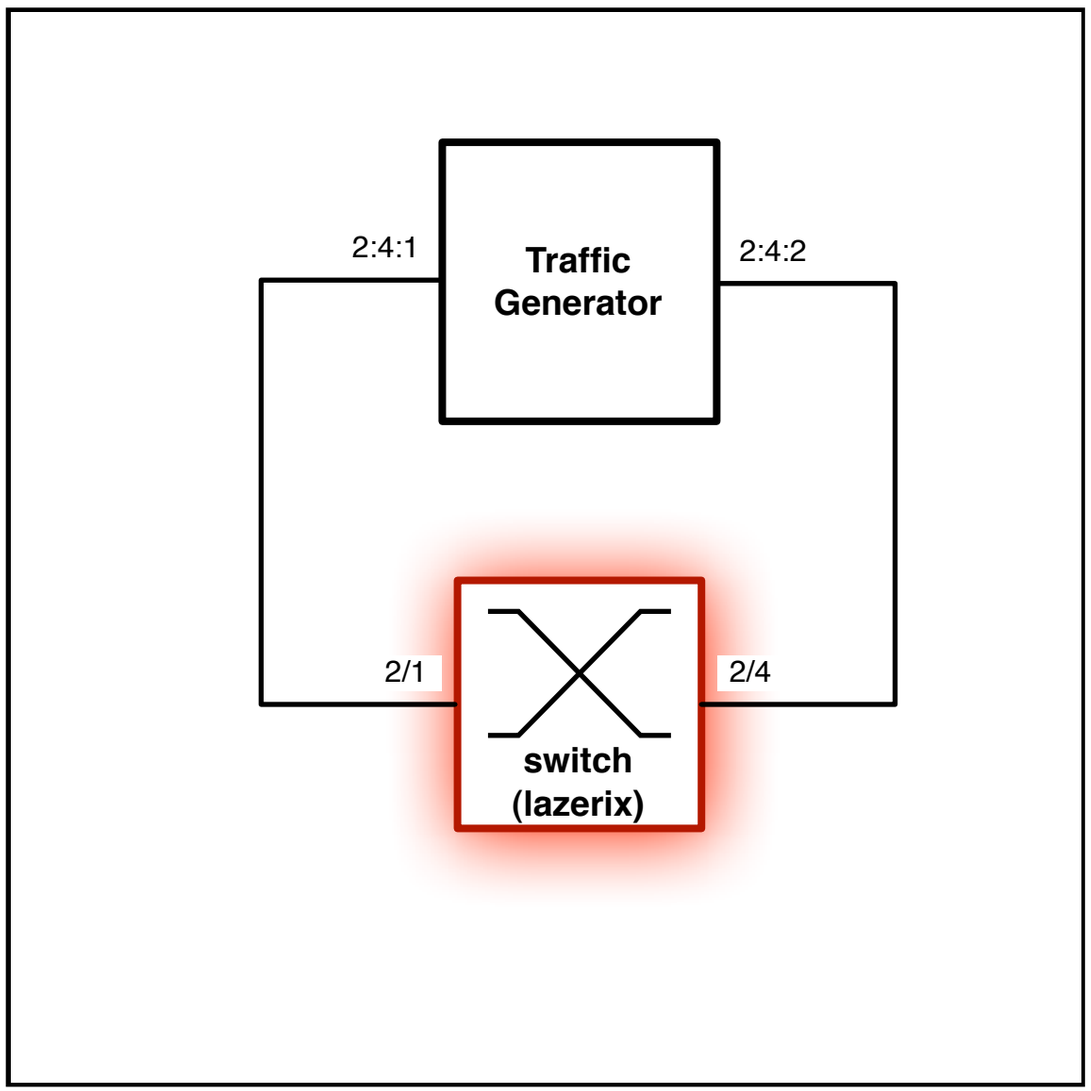
Ariën Vijn
arien.vijn@ams-ix.net

Demo Setup



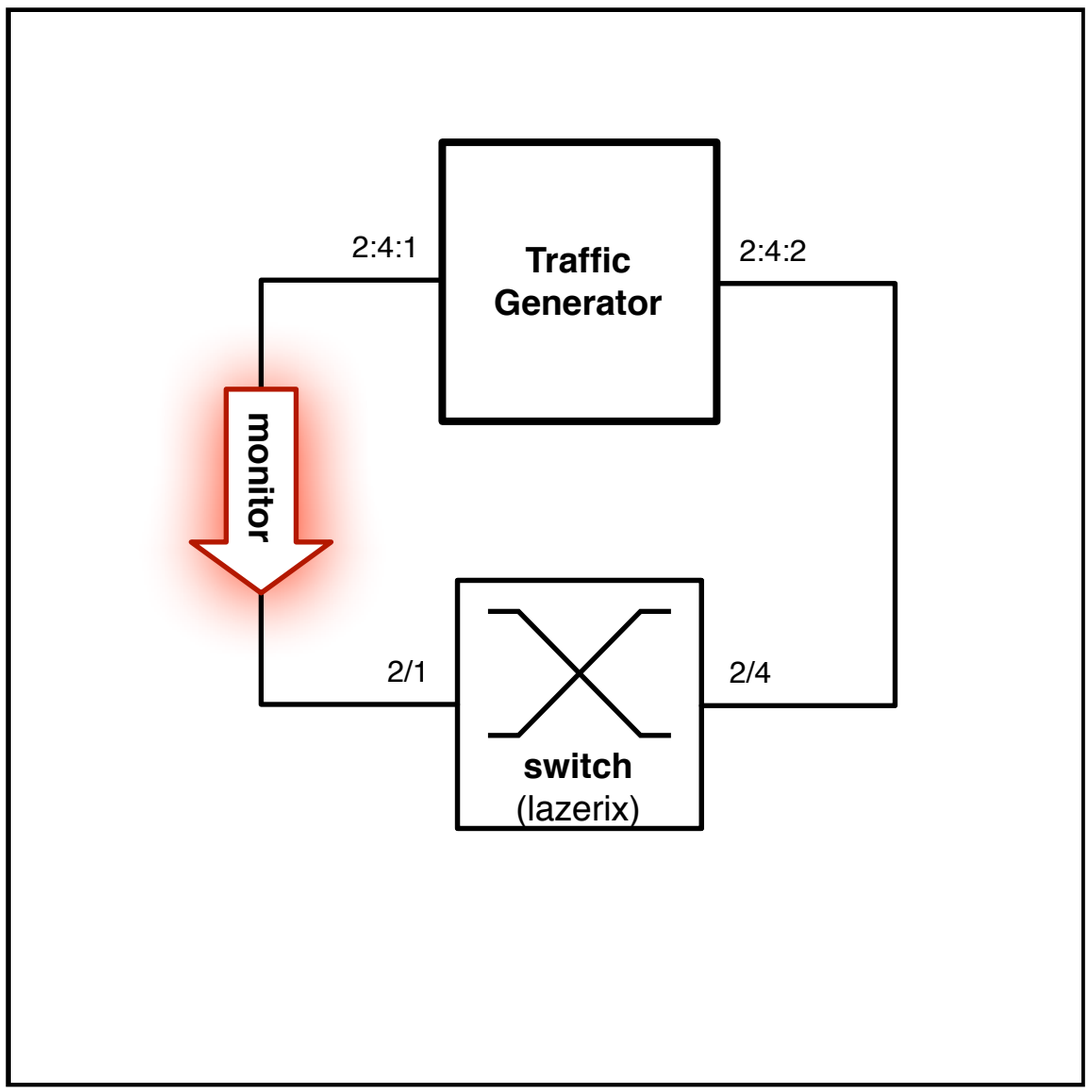
- Traffic generator
  - Screen with Windows interface

Demo Setup



- Traffic generator
  - Screen with Windows interface.
- Ethernet switch called “lazerix”
- One of the terminal window (look at prompt)

Demo Setup



- Traffic generator
  - Screen with Windows interface
- Ethernet switch called “lazerix”
  - One of the terminal window (look at prompt)
- We’ll look at the traffic on the wire from 2:4:1 to 2/1
  - Monitorbox called “hysterix”

Let’s have a look

10GbE monitoring
(some slides now)

Ariën Vijn
arien.vijn@ams-ix.net

Agenda

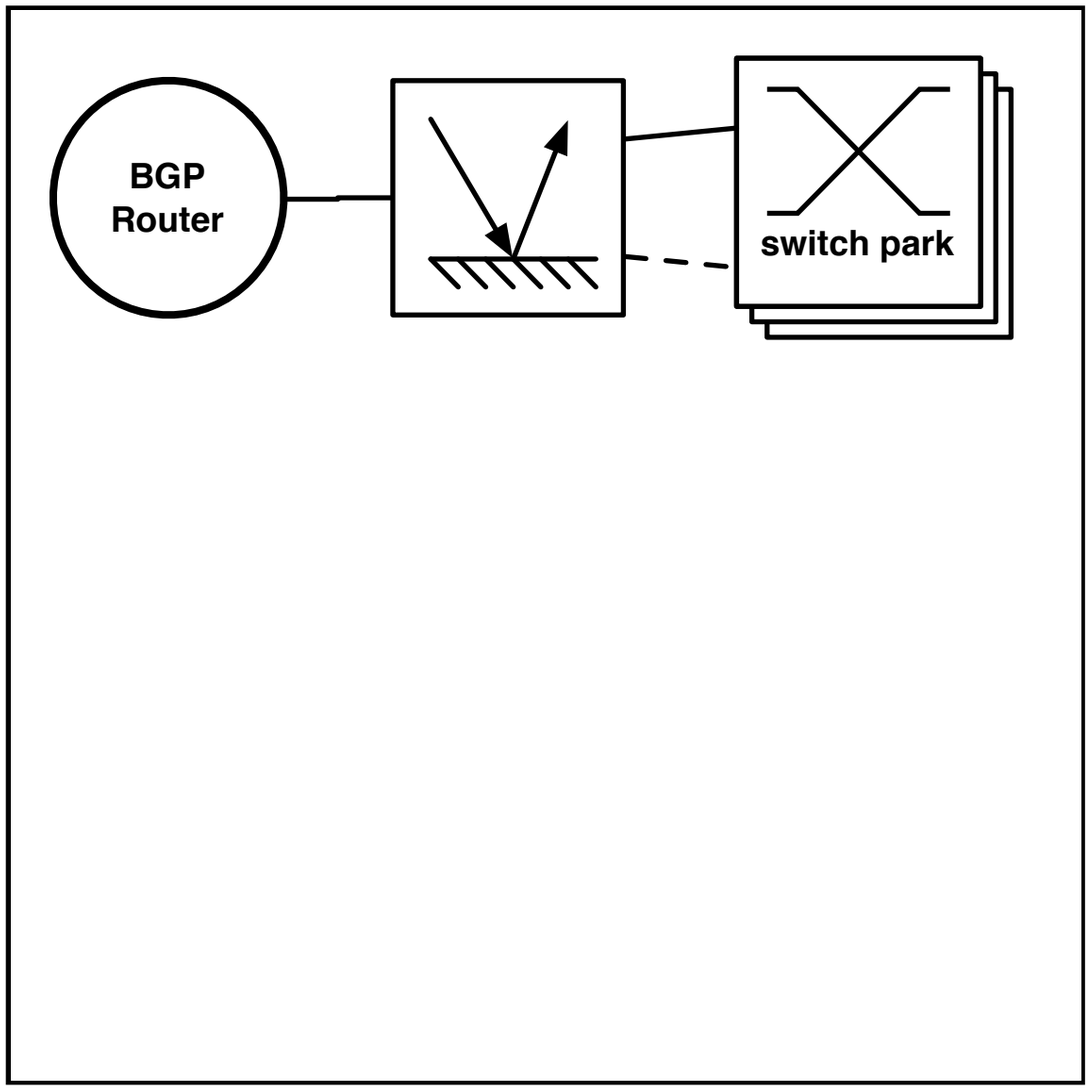
- An example.
- The problems with passive 10GbE monitoring.
  - Performance of generic computer hardware.
  - Issues with the mirror port feature.
- Overview of our solution.
  - Hardware architecture.
  - Homegrown firmware and software.
- A bit more detail.
- More to demo.

Agenda

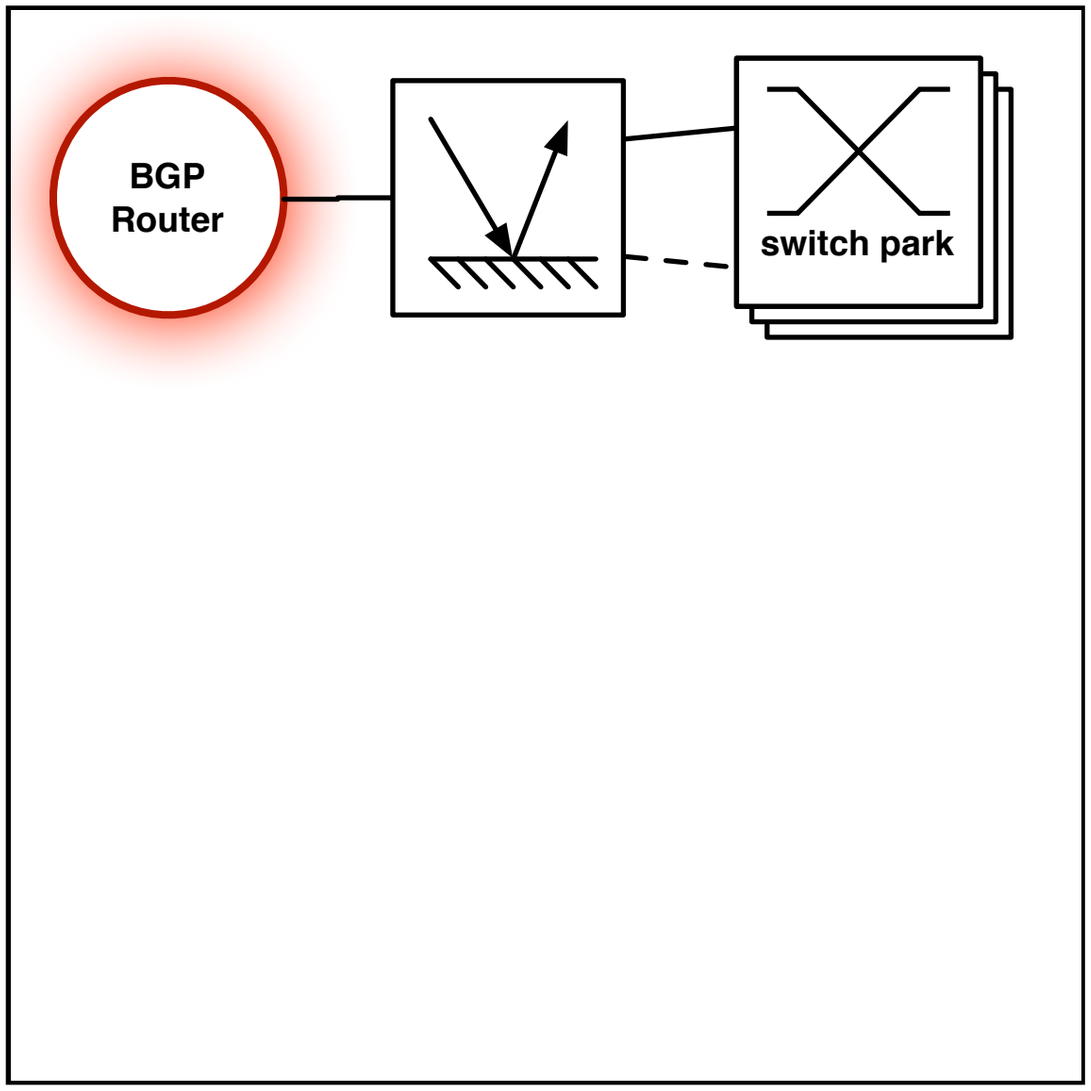
- An example.
- The problems with passive 10GbE monitoring.
  - Performance of generic computer hardware.
  - Issues with the mirror port feature.
- Overview of our solution.
  - Hardware architecture.
  - Homegrown firmware and software.
- A bit more detail.
- More to demo.

Example

- Normal situation

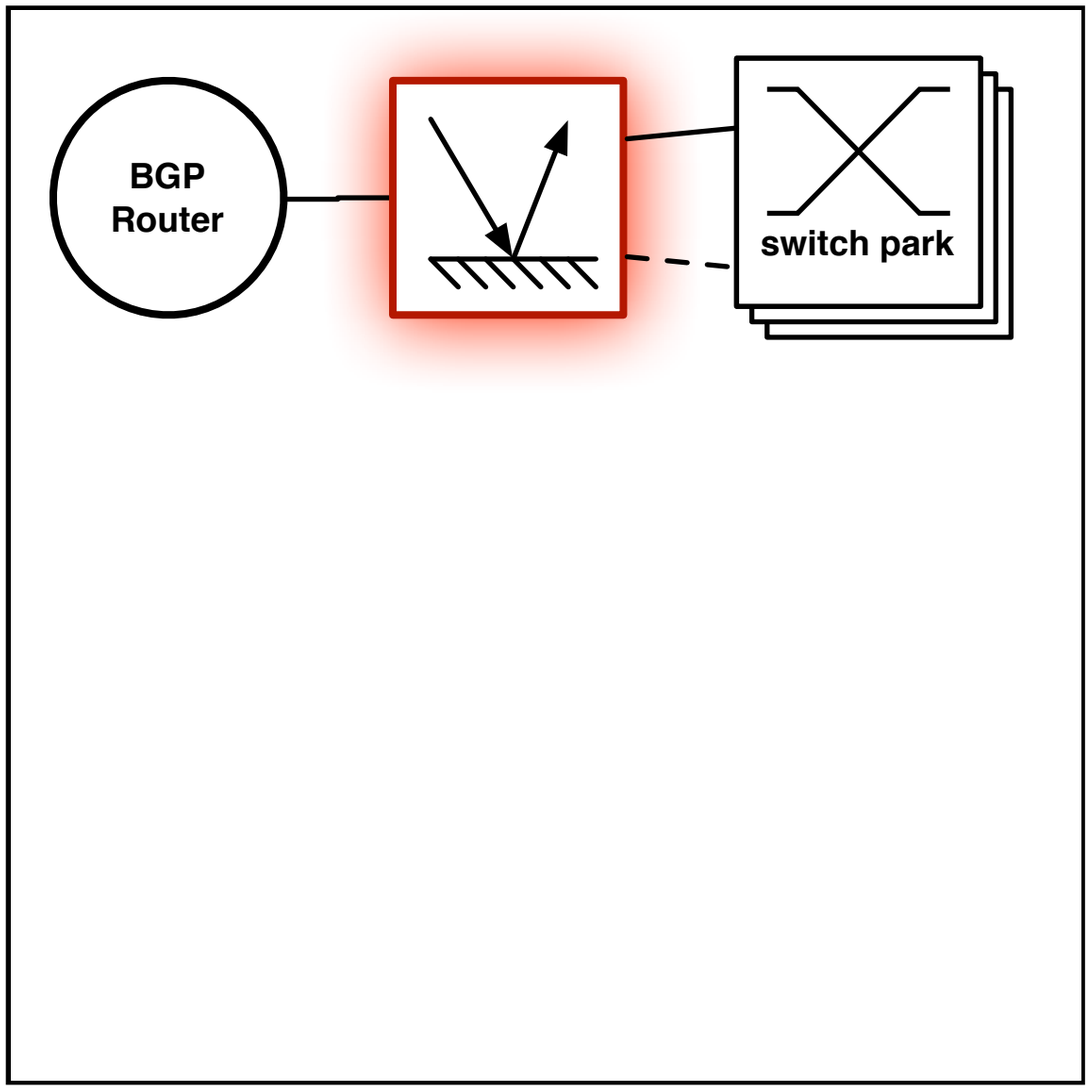


Example



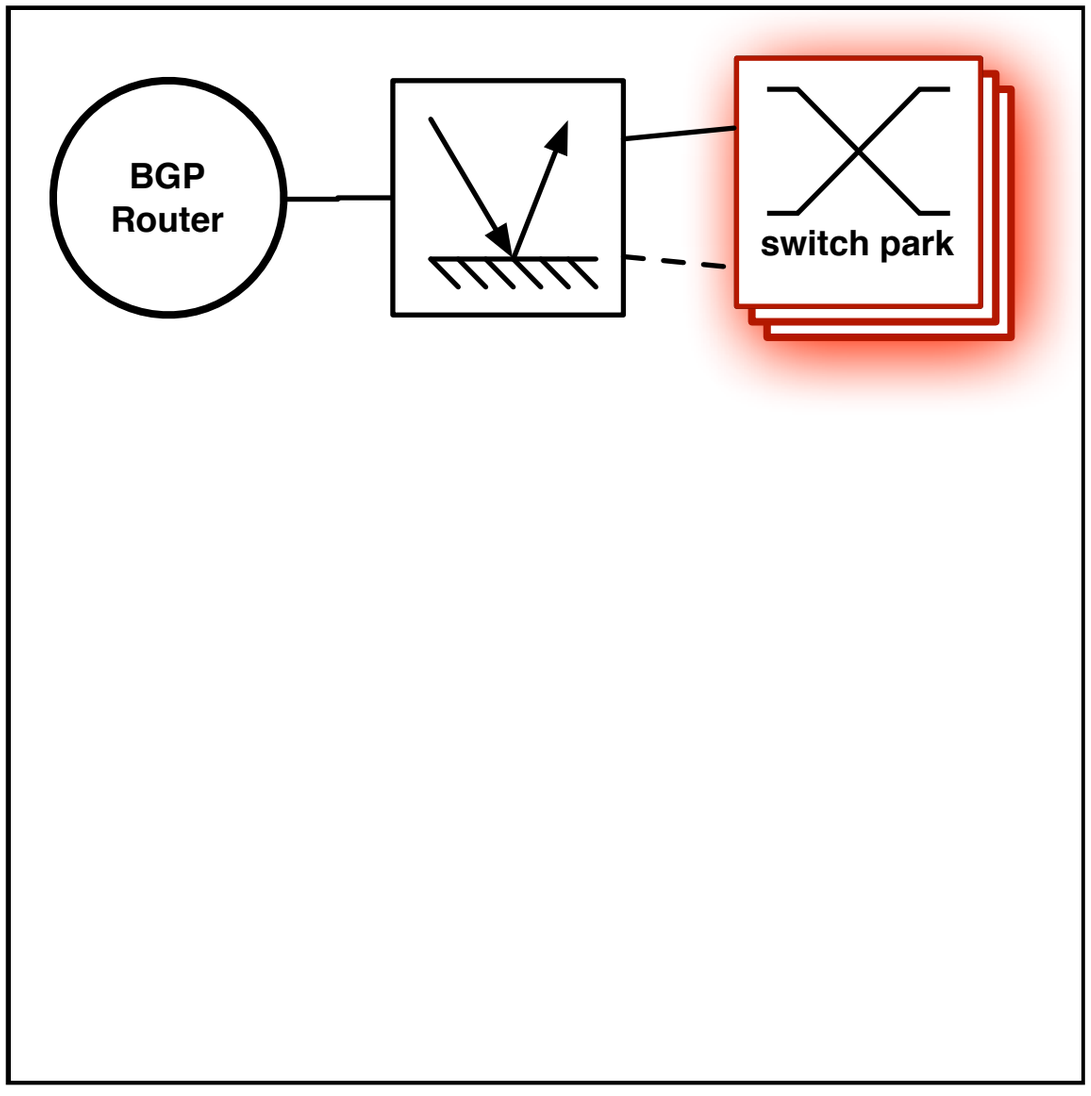
- Normal situation
  - 10GbE member router

Example



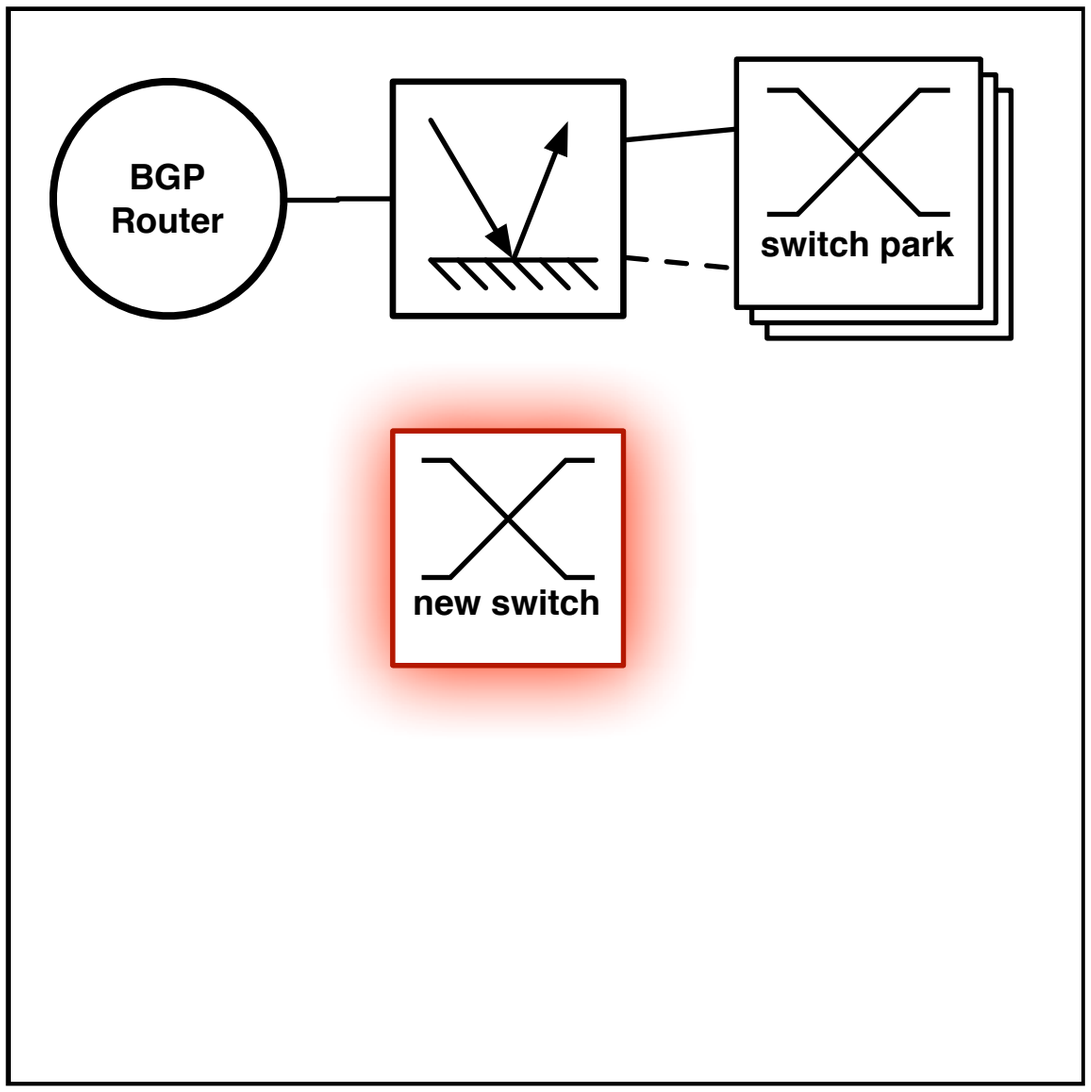
- Normal situation
  - 10GbE member router
  - Photonic Cross Connect (PXC)

Example



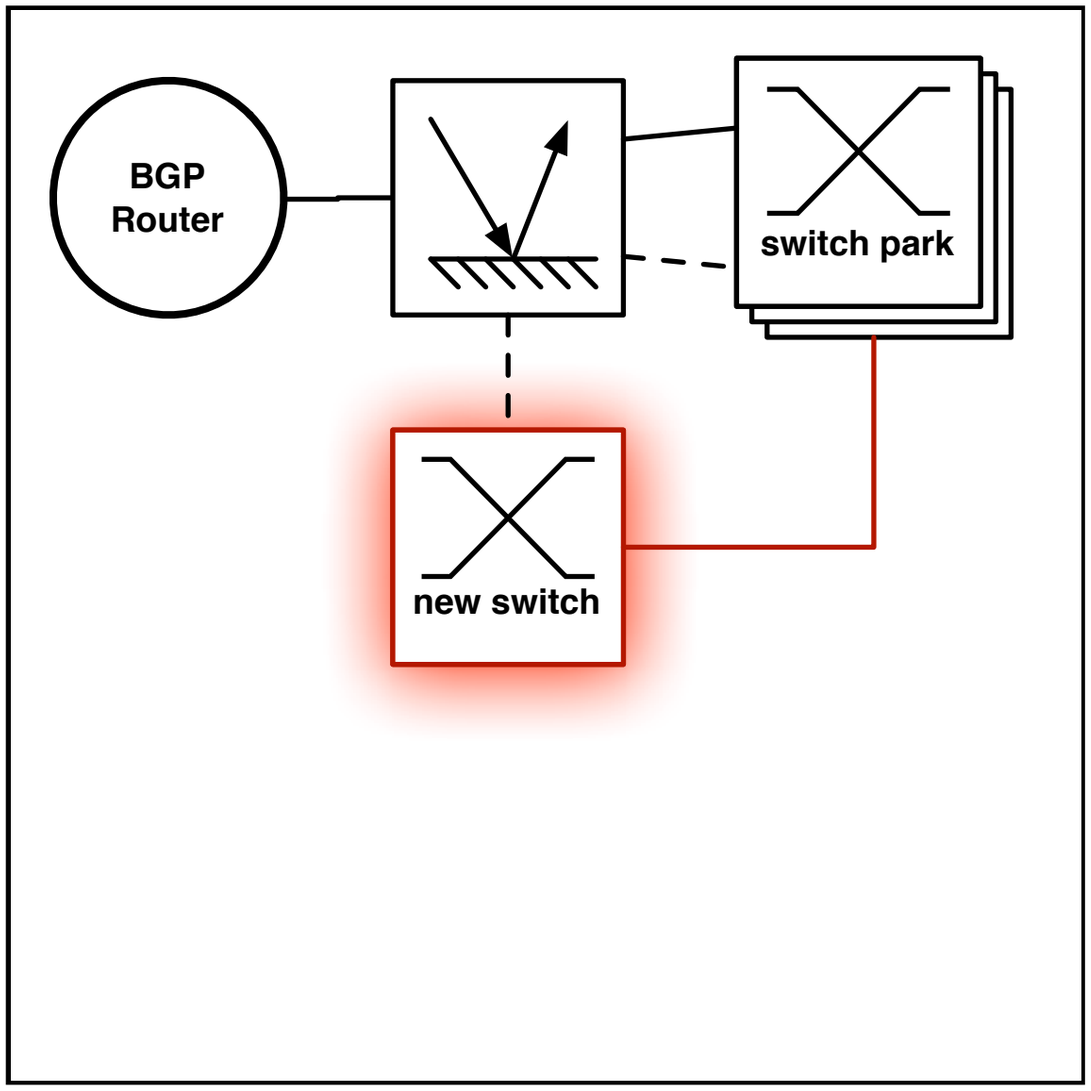
- Normal situation
  - 10GbE member router
  - Photonic Cross Connect (PXC)
  - Redundant switch park

Example



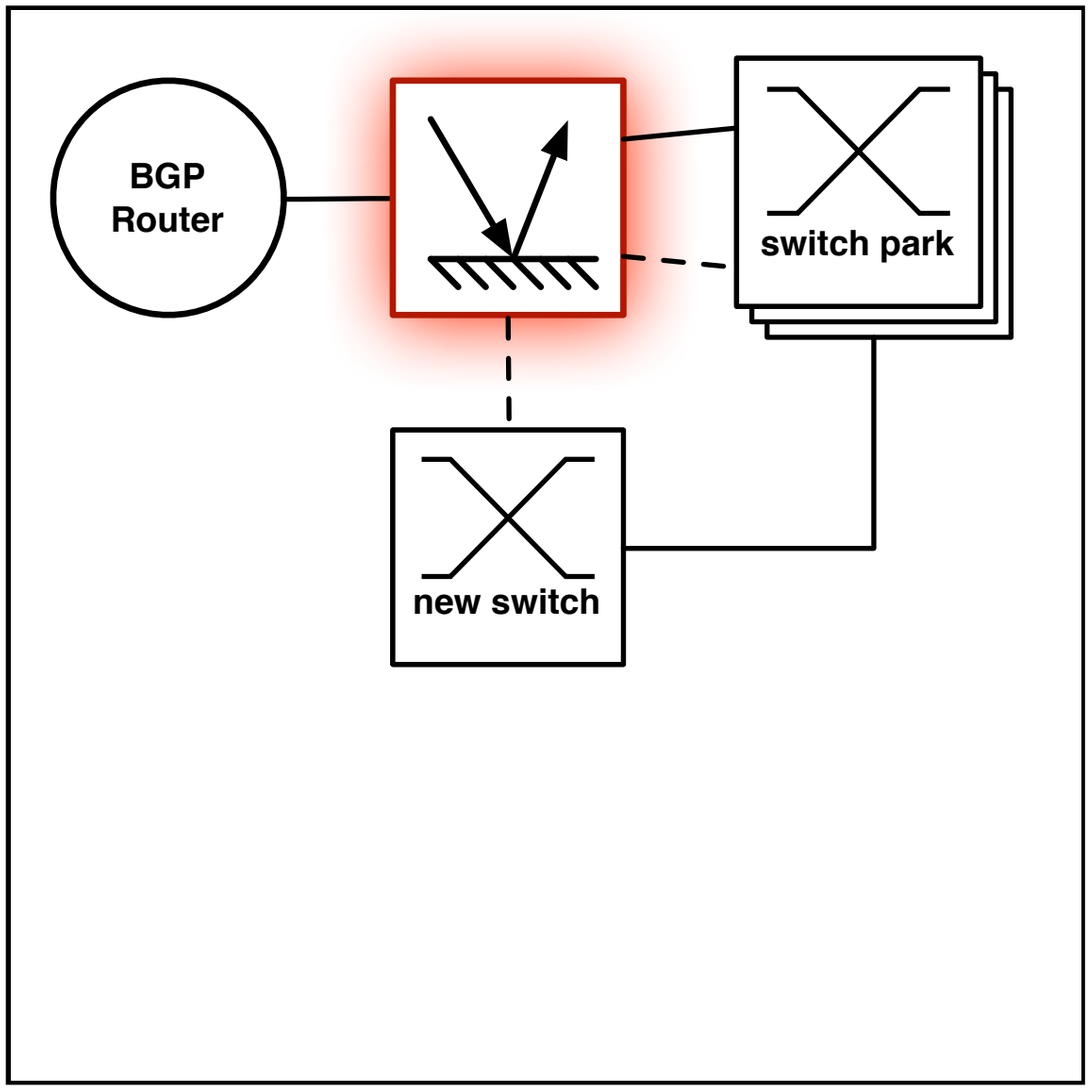
- Normal situation
  - 10GbE member router
  - Photonic Cross Connect (PXC)
  - Redundant switch park
- New switch

Example



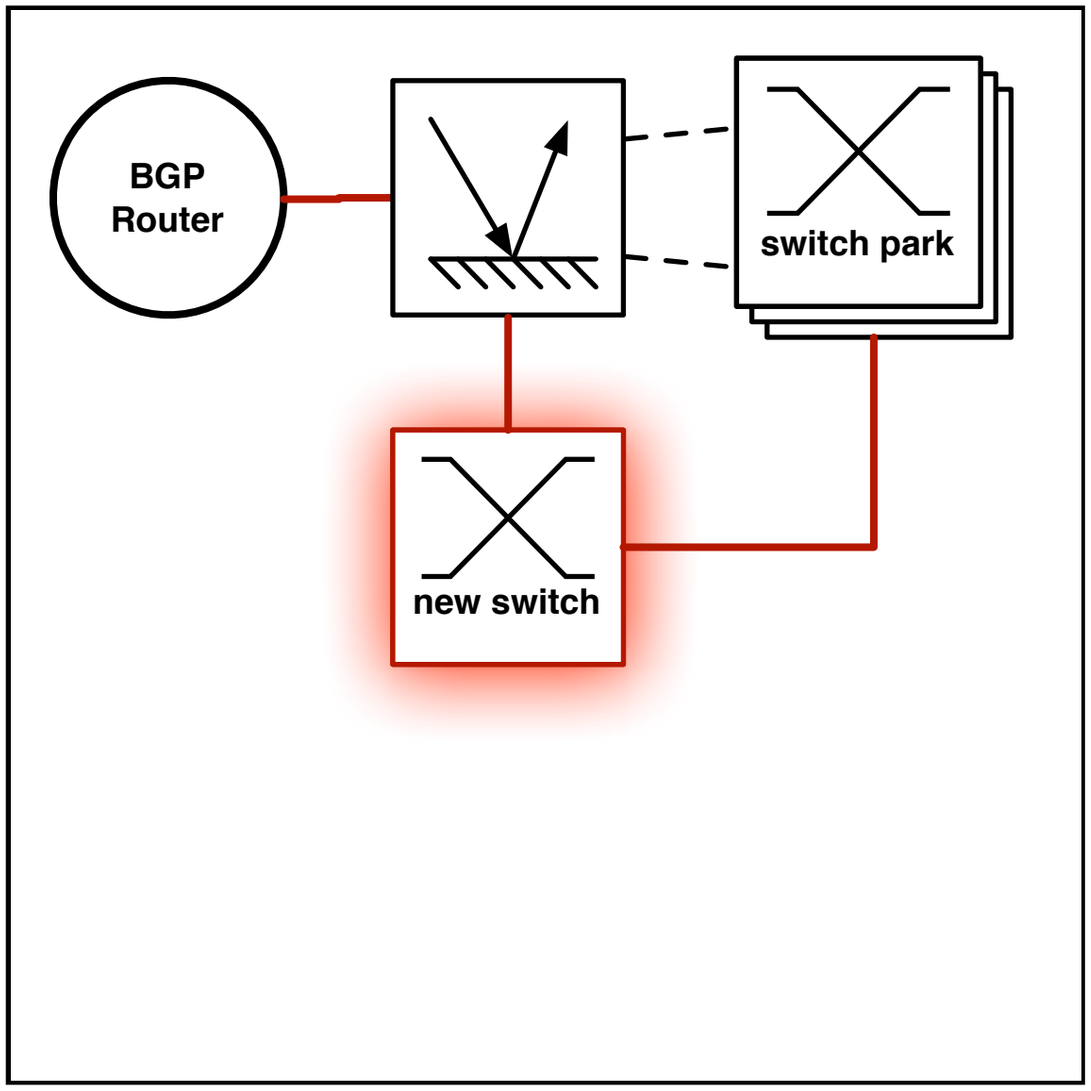
- Normal situation
  - 10GbE member router
  - Photonic Cross Connect (PXC)
  - Redundant switch park
- New switch
  - Connected to the switch park

Example



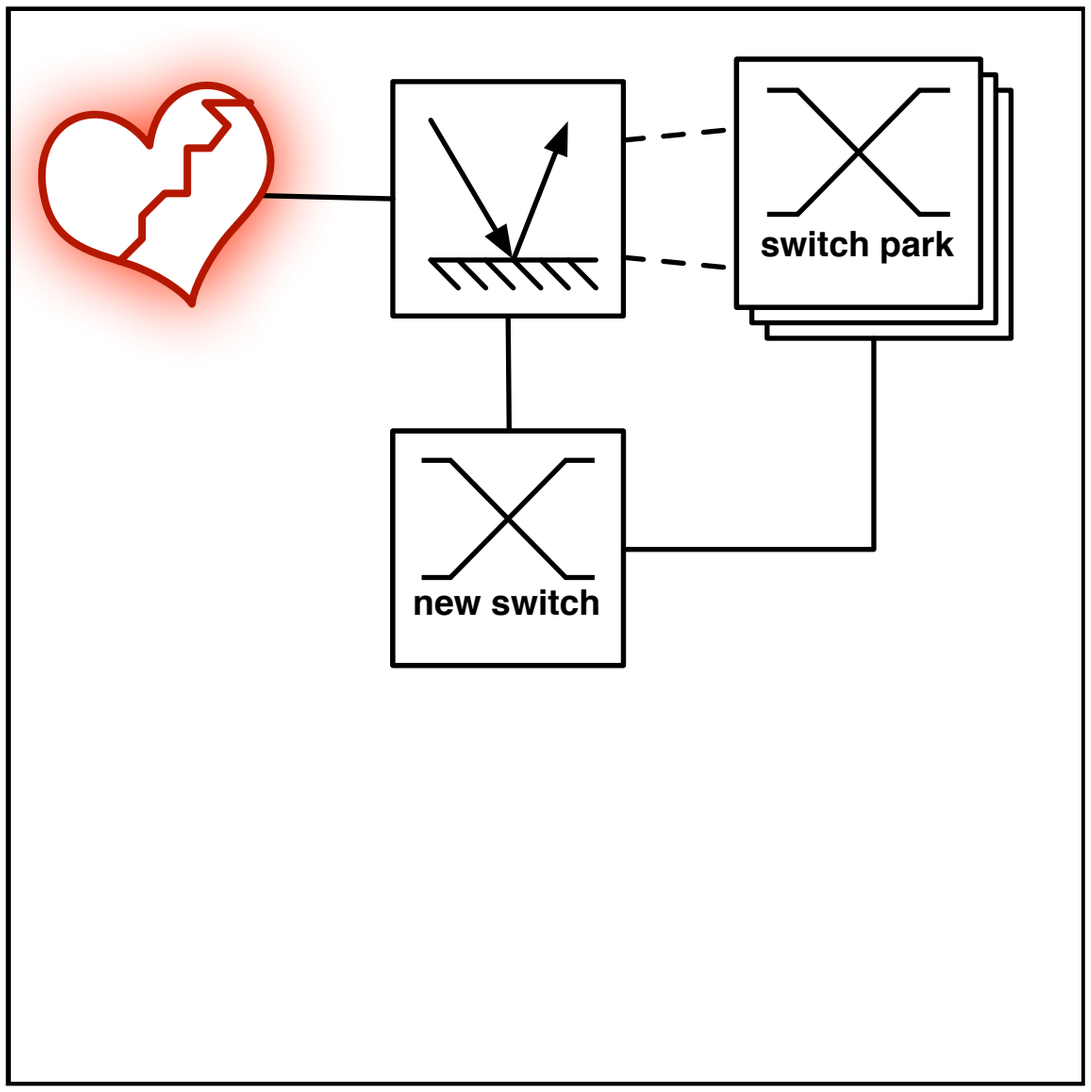
- Normal situation
  - 10GbE member router
  - Photonic Cross Connect (PXC)
  - Redundant switch park
- New switch
  - Connected to the switch park

Example



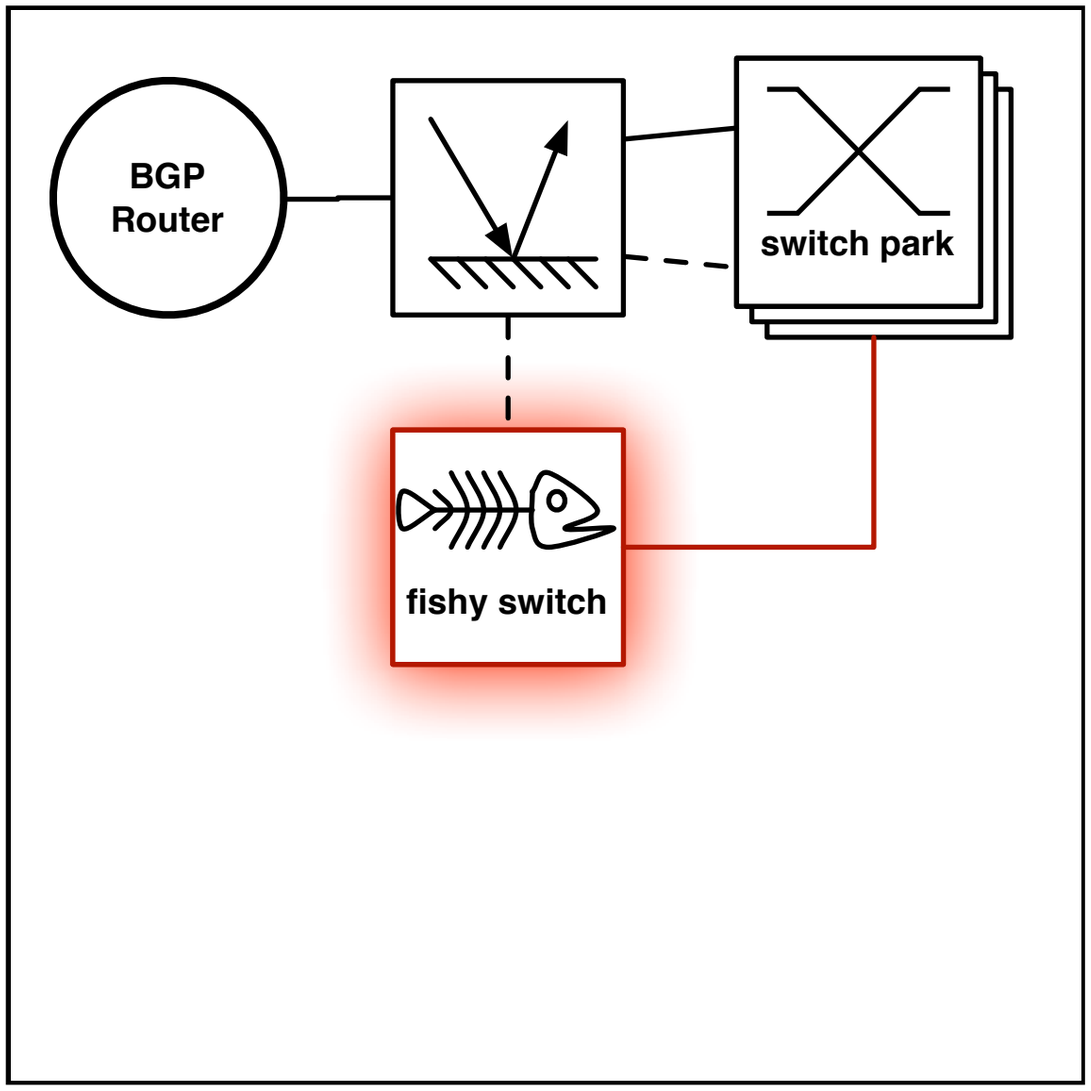
- Normal situation
  - 10GbE member router
  - Photonic Cross Connect (PXC)
  - Redundant switch park
- New switch
  - Connected to the switch park
  - Moved in production

Example



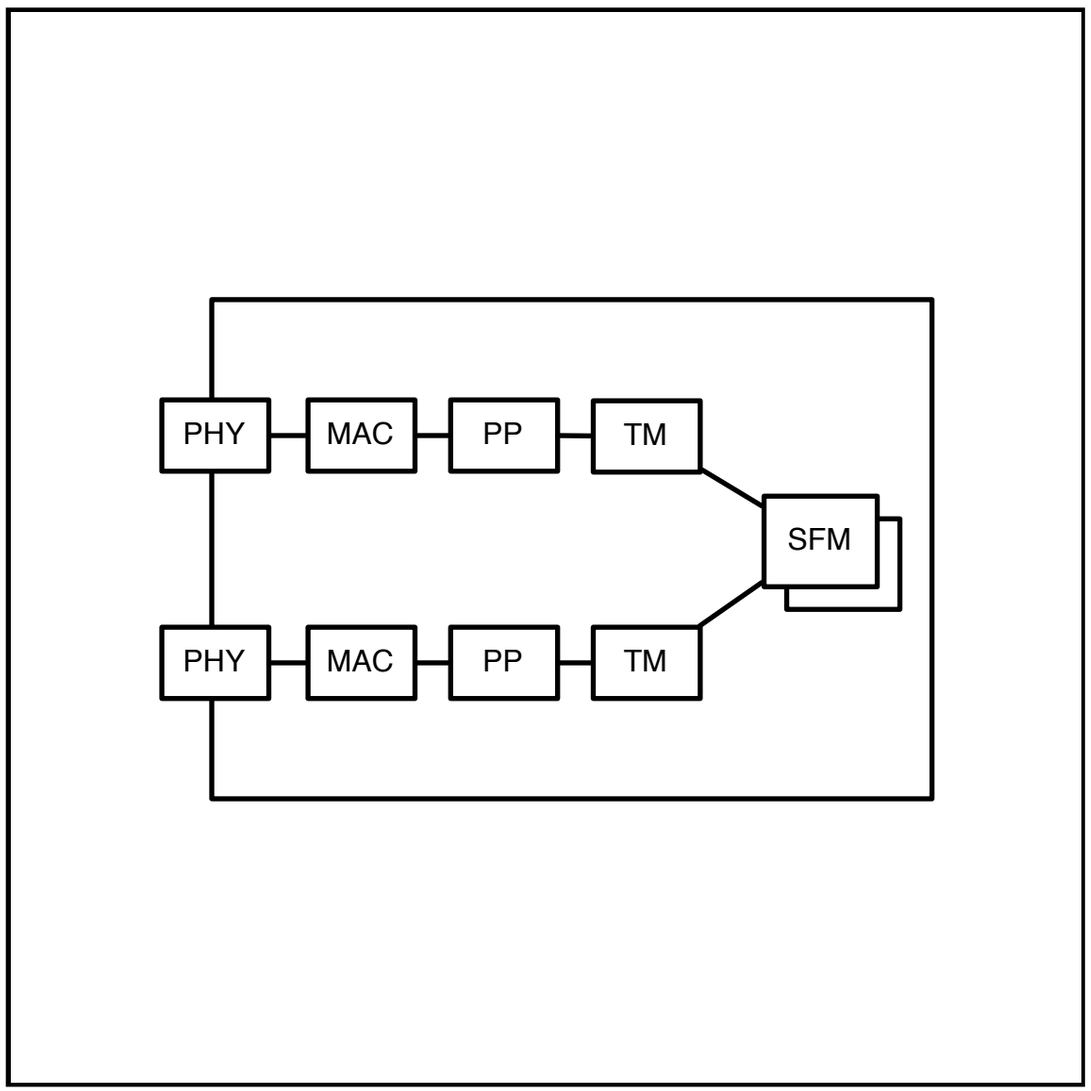
- Normal situation
  - 10GbE member router
  - Photonic Cross Connect (PXC)
  - Redundant switch park
- New switch
  - Connected to the switch park
  - Moved in production
  - Member feels unhappy

Example



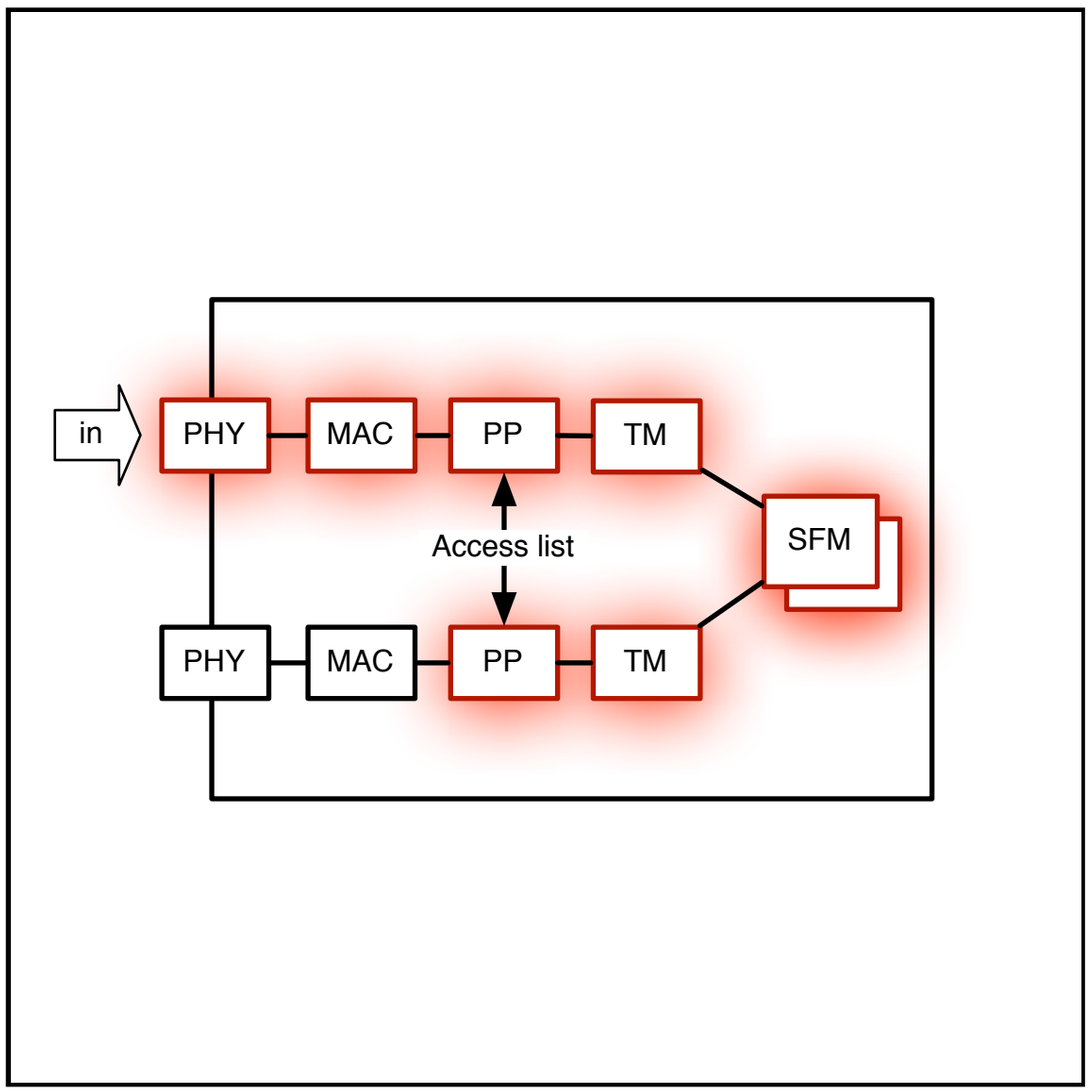
- Normal situation
  - 10GbE member router
  - Photonic Cross Connect (PXC)
  - Redundant switch park
- New switch
  - Connected to the switch park
  - Moved in production
  - Member feels unhappy
  - Something must be wrong with the new switch

Example



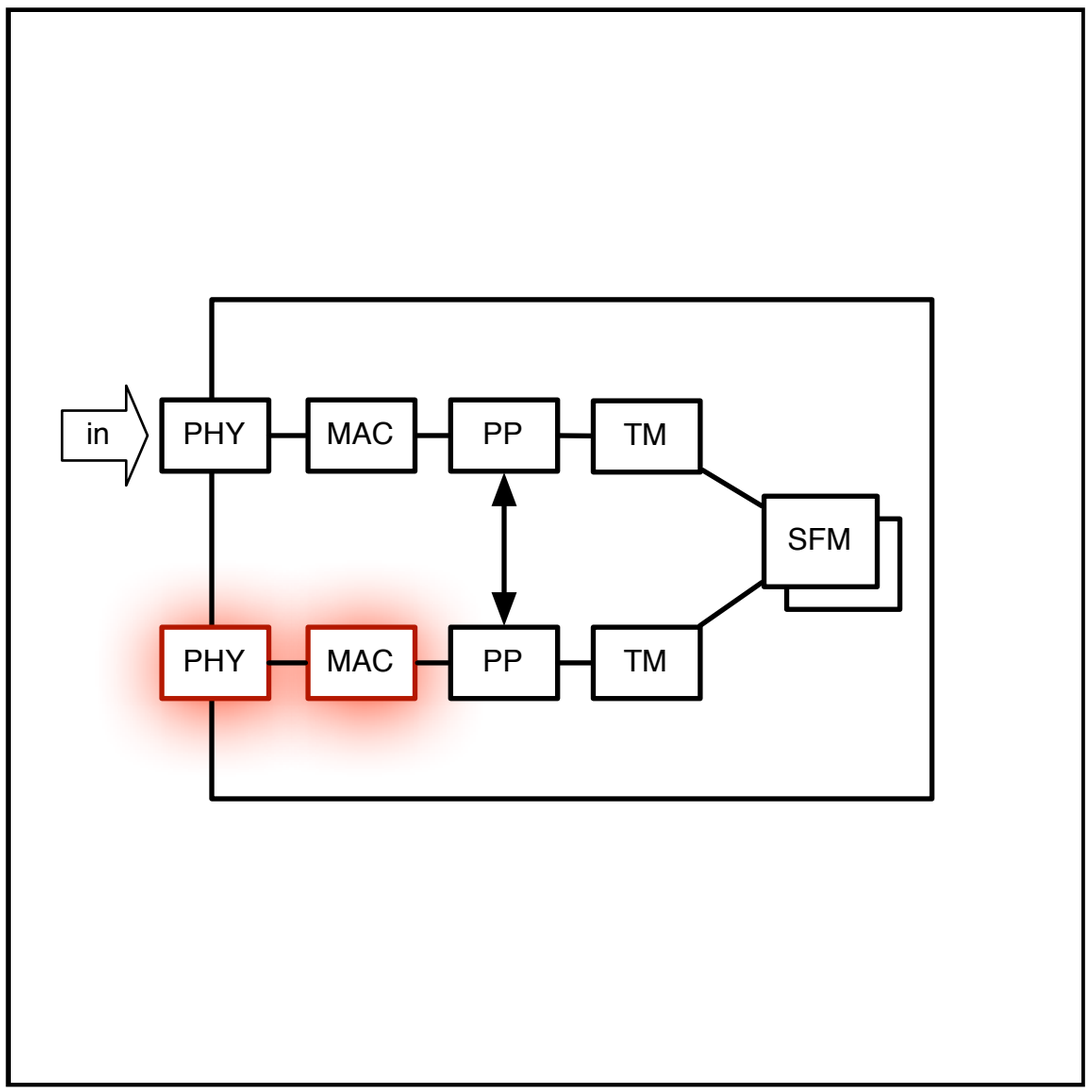
- Debugging the switch (the vendor's way)

Example



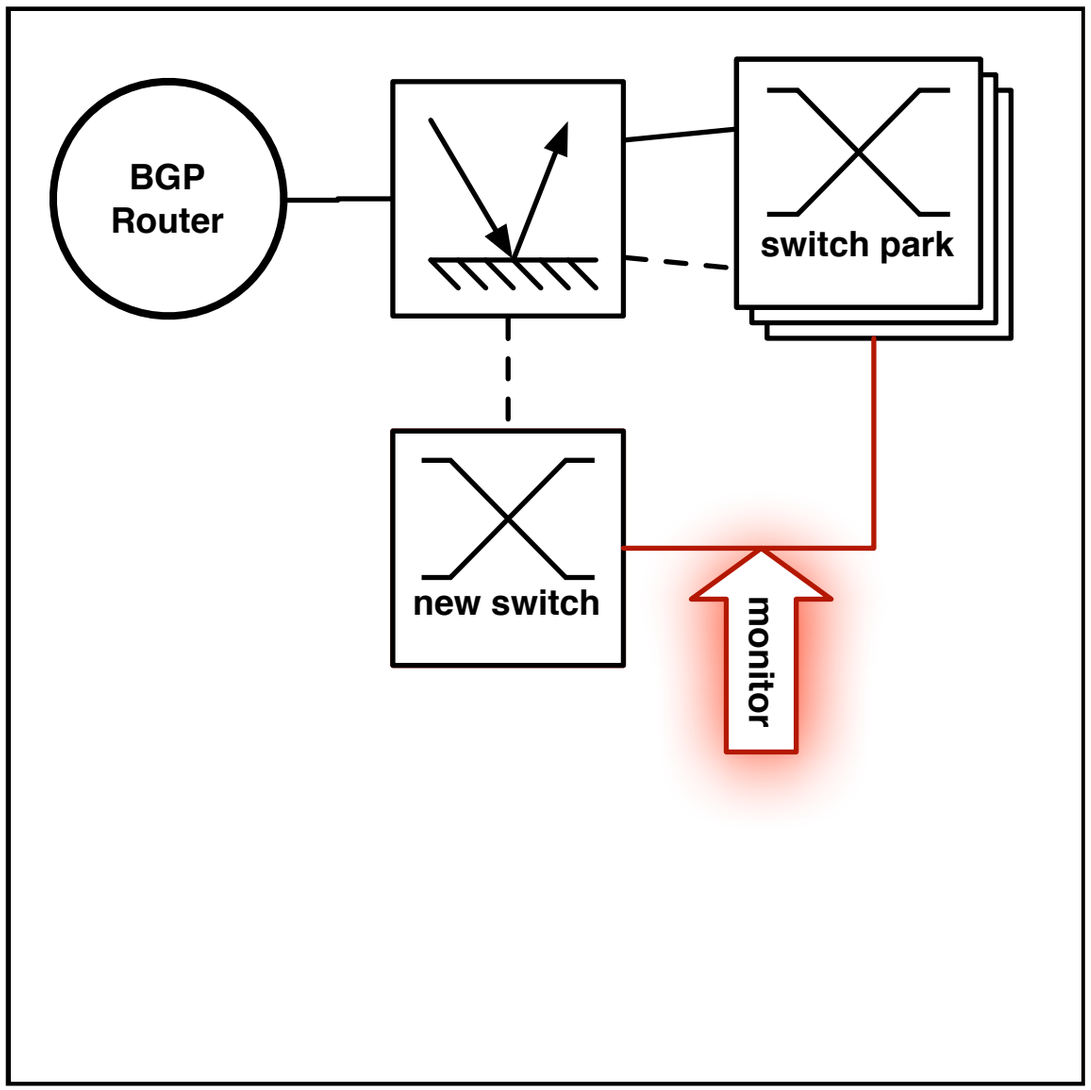
- Debugging the switch (the vendor's way)
  - Layer-2 access list accounting

Example



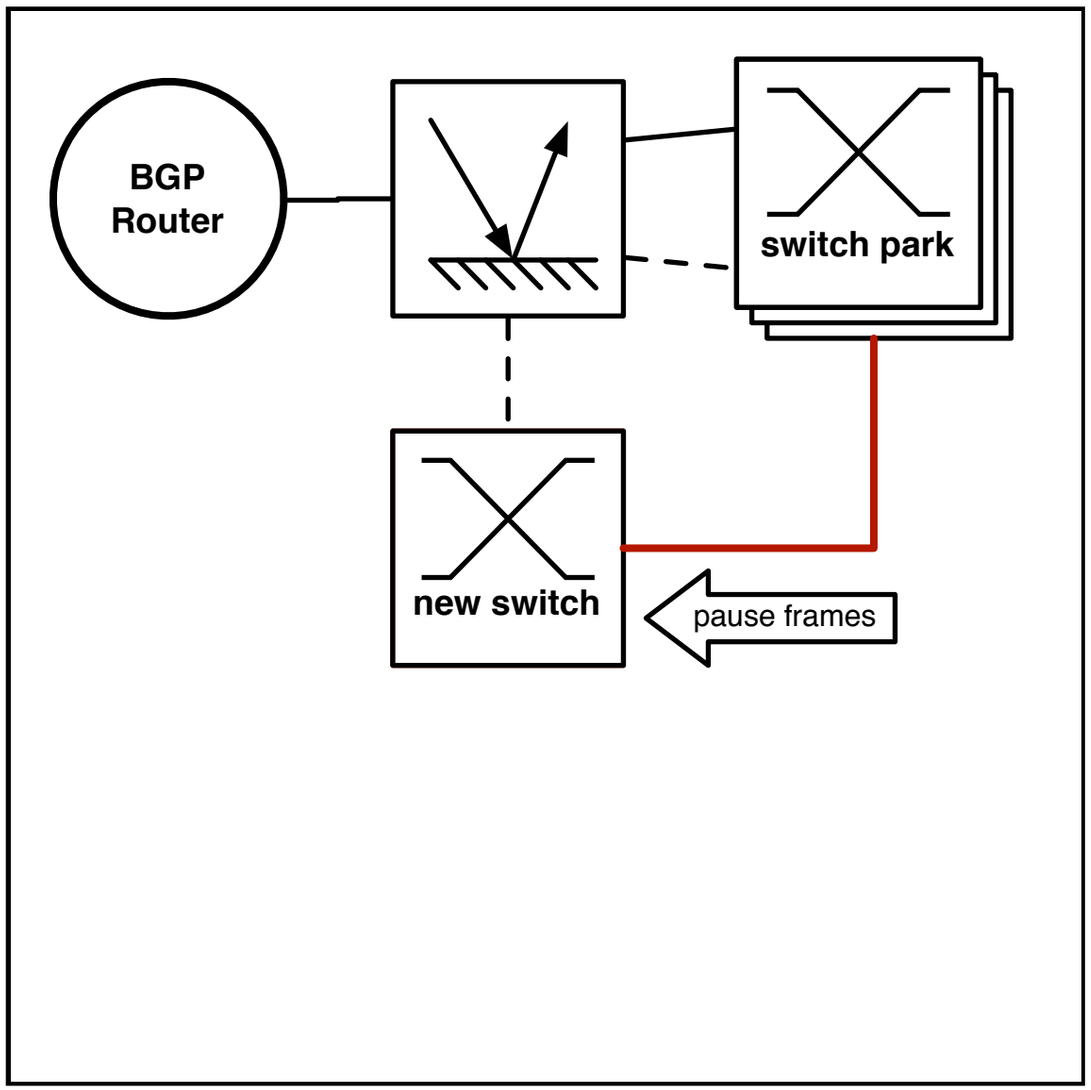
- Debugging the switch (the vendor's way)
  - Layer-2 access list accounting
  - Misses part of the egress path
  - "Cannot be the problem because the same chips serve the ingress path."
  - This took weeks

Example



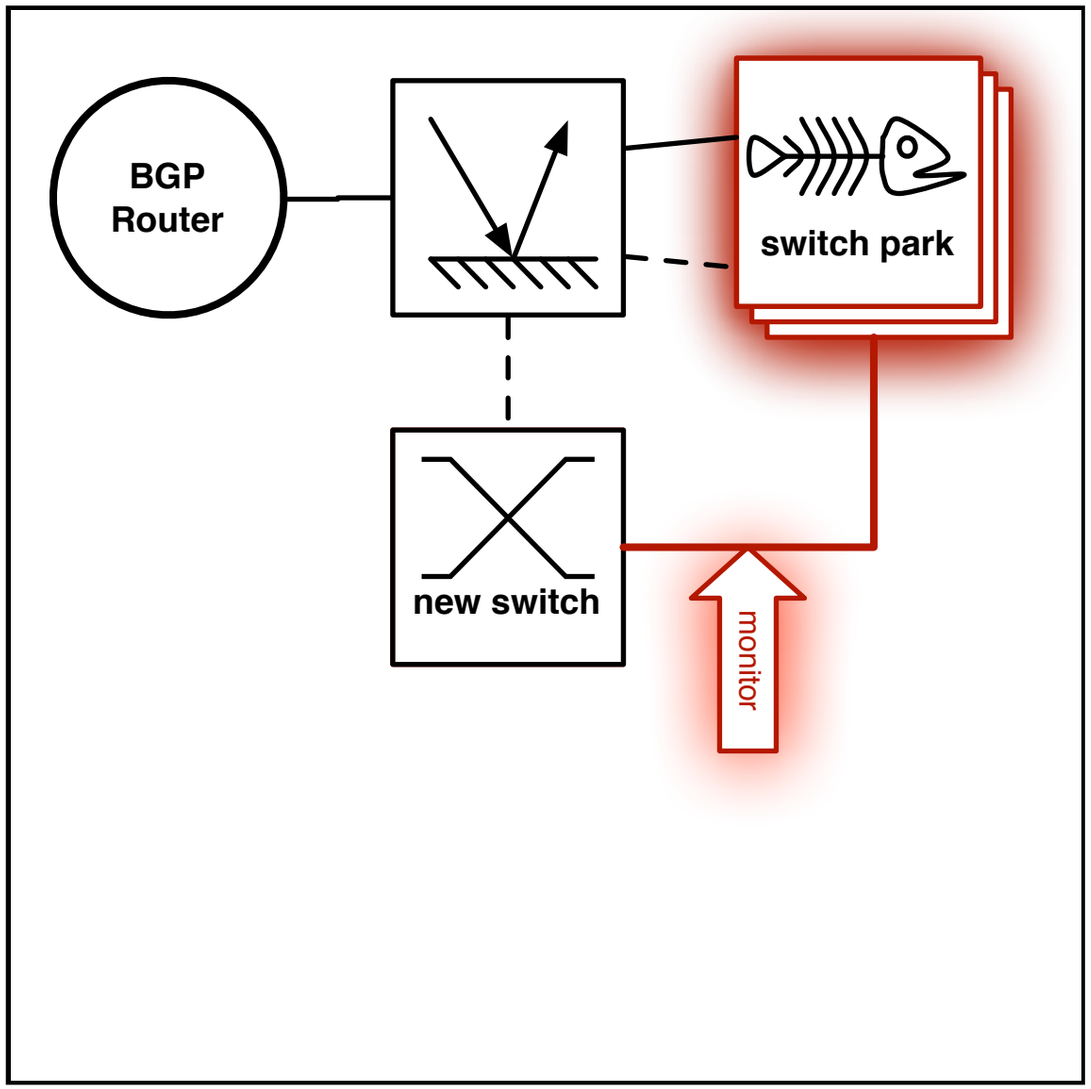
- Debugging the network (our way)
  - Passive monitoring
  - Watching the traffic as it passes by.

Example



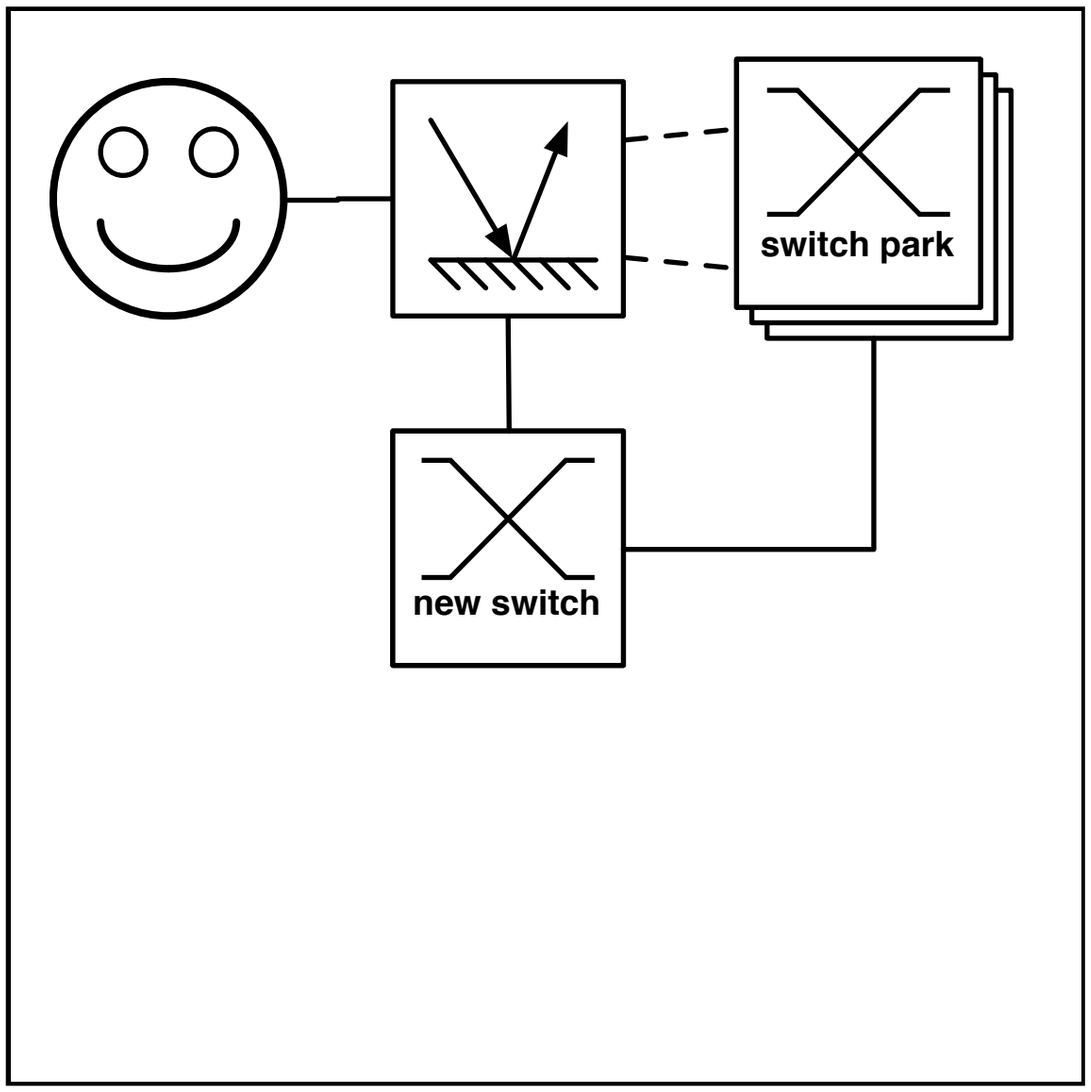
- Debugging the network (our way)
  - Passive monitoring
  - Watching the traffic as it passes by.
- What did we see?
  - A lot of incoming pause frames!

Example



- Debugging the network (our way)
  - Passive monitoring
  - Watching the traffic as it passes by.
- What did we see?
  - A lot of incoming pause frames!
  - We had a problem in our switch park!
  - Issue found within two minutes after invoking the monitor system.

Example

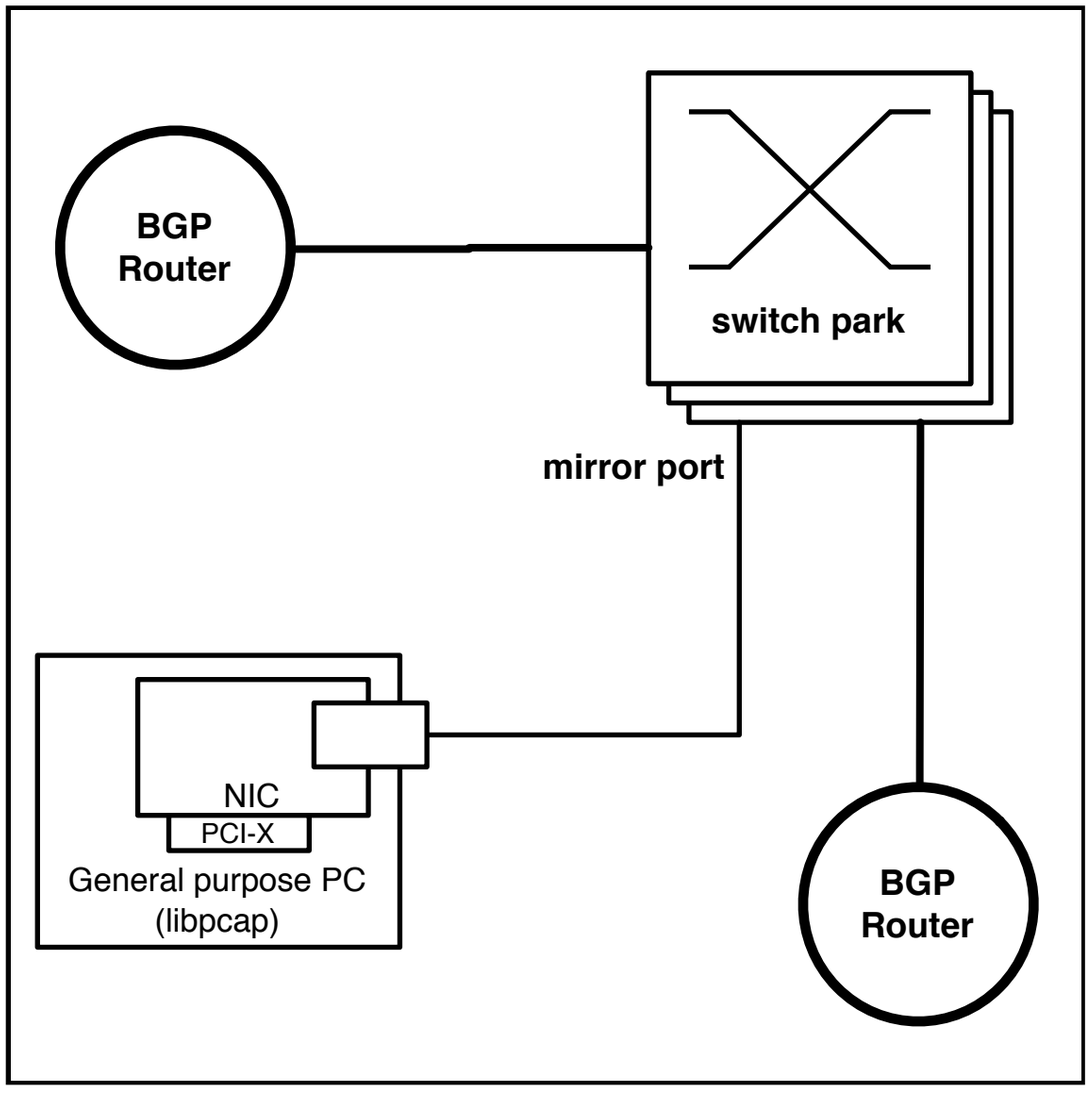


- Debugging the network (our way)
  - Passive monitoring
  - Watching the traffic as it passes by.
- What did we see?
  - A lot of incoming pause frames!
  - We had a problem in our switch park!
  - This took 2 minutes after the monitor system was installed.
  - Software upgrade solved the problem.

Agenda

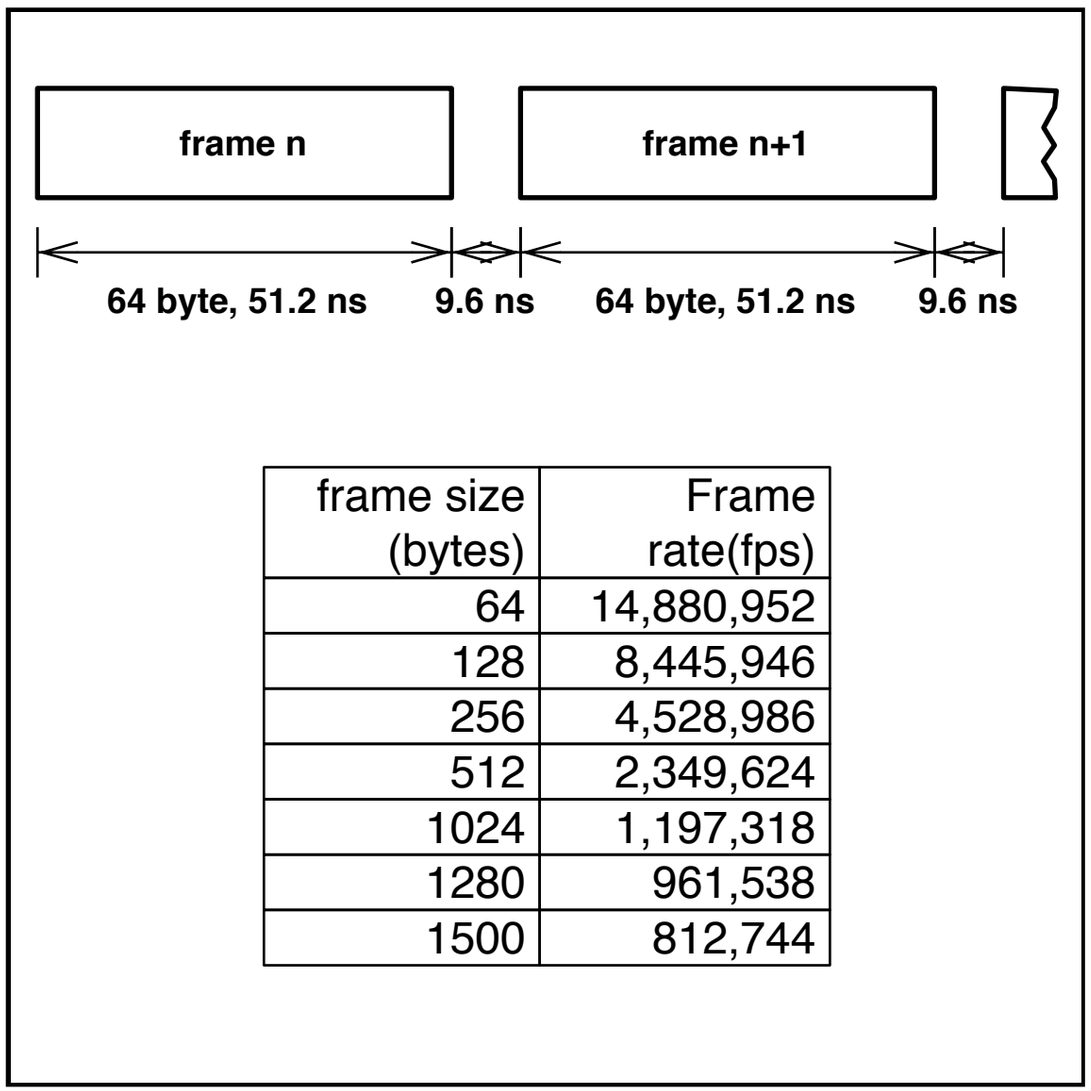
- An example.
- The problems with passive 10GbE monitoring.
  - Performance of generic computer hardware.
  - Issues with the mirror port feature.
- Overview of our solution.
  - Hardware architecture.
  - Homegrown firmware and software.
- A bit more detail.
- More to demo.

Common setup



- Mirror port feature
- General purpose PC
  - Tools based on libpcap

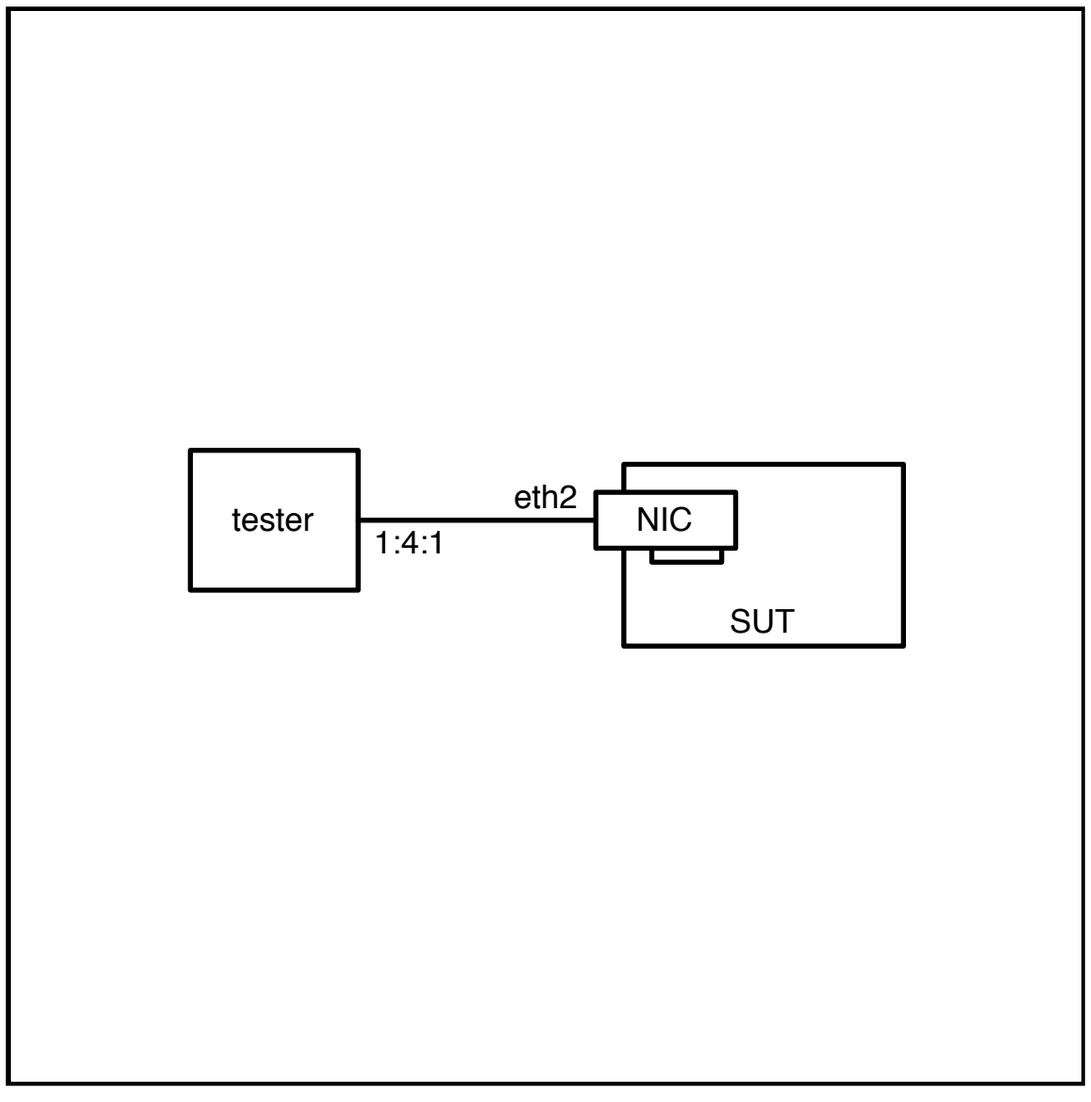
Generic purpose PC



frame size (bytes)	Frame rate(fps)
64	14,880,952
128	8,445,946
256	4,528,986
512	2,349,624
1024	1,197,318
1280	961,538
1500	812,744

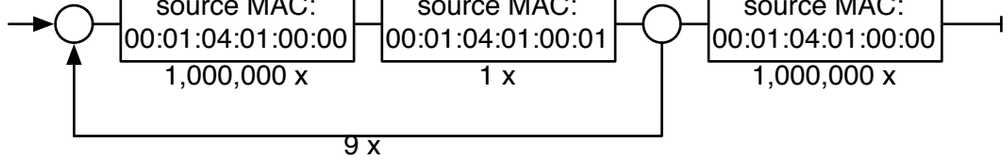
- 10GbE frame rates
  - 14.8 million frames per seconnd

Generic purpose PC



- 10GbE frame rates
  - 14.8 million frames per second
- Test setup
  - Traffic generator to 'normal' Intel 10GbE NIC

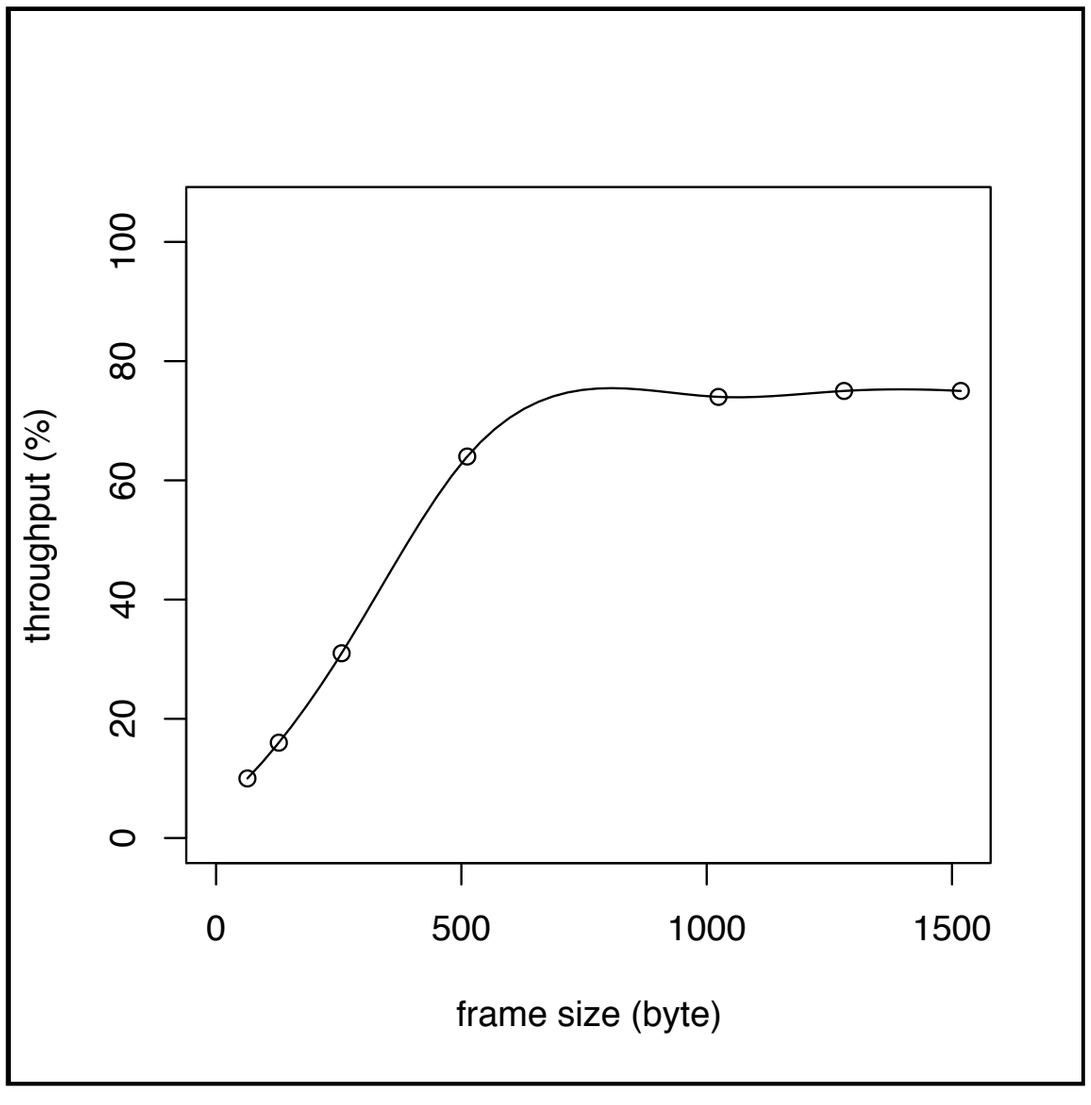
Generic purpose PC



frame size (bytes)	bandwidth (%)	Frame rate(Mfps)
64	10	1.488
128	16	1.351
256	31	1.404
512	64	1.504
1024	74	0.886
1280	75	0.721
1500	75	0.609

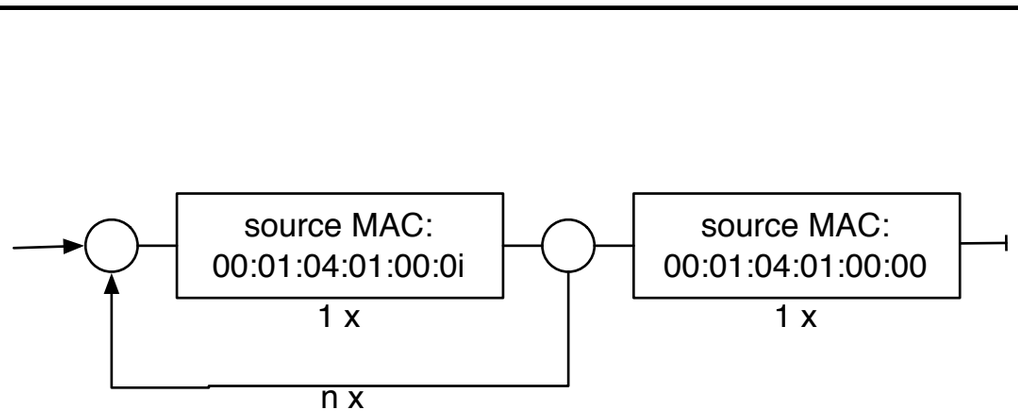
- 10GbE frame rates
  - 14.8 million frames per second
- Test setup
  - Traffic generator to 'normal' Intel 10GbE NIC
- One out of one million.
  - Reduced data rate until all frames were captured
  - Note the more or less constant frame rate of 1.5Mfps
  - PCI-X bus speed harms larger sized frame.

Generic purpose PC



- 10GbE frame rates
  - 14.8 million frames per second
- Test setup
  - Traffic generator to 'normal' Intel 10GbE NIC
- One out of one million frames
  - Reduced data rate until all frames were captured
  - Note the more or less constant frame rate of 1.5Mfps
  - PCI-X bus speed harms larger frame sizes.

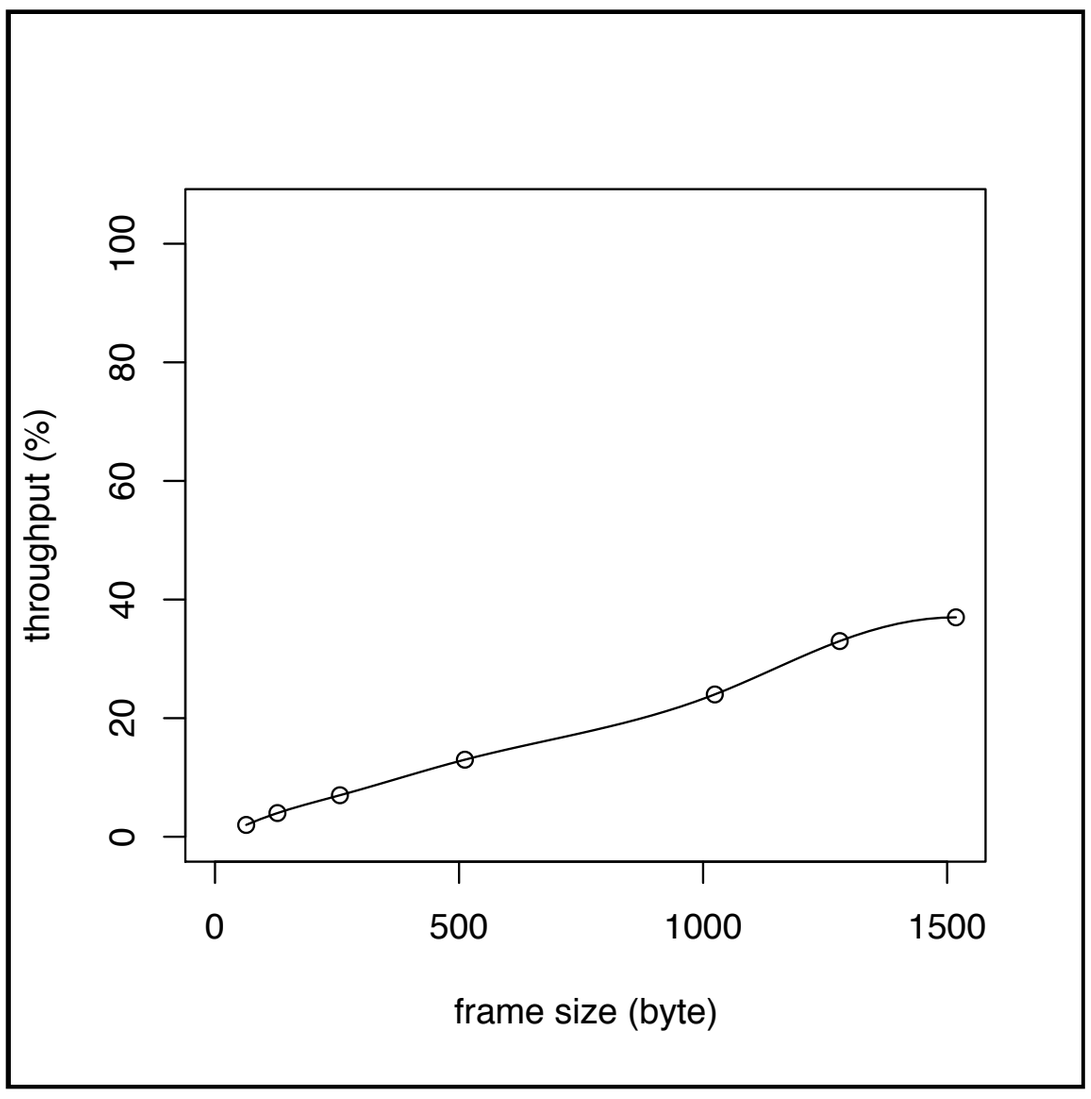
Generic purpose PC



frame size (bytes)	bandwidth (%)	Frame rate(kfps)
64	2	298
128	4	338
256	7	317
512	13	305
1024	24	287
1280	33	317
1500	37	301

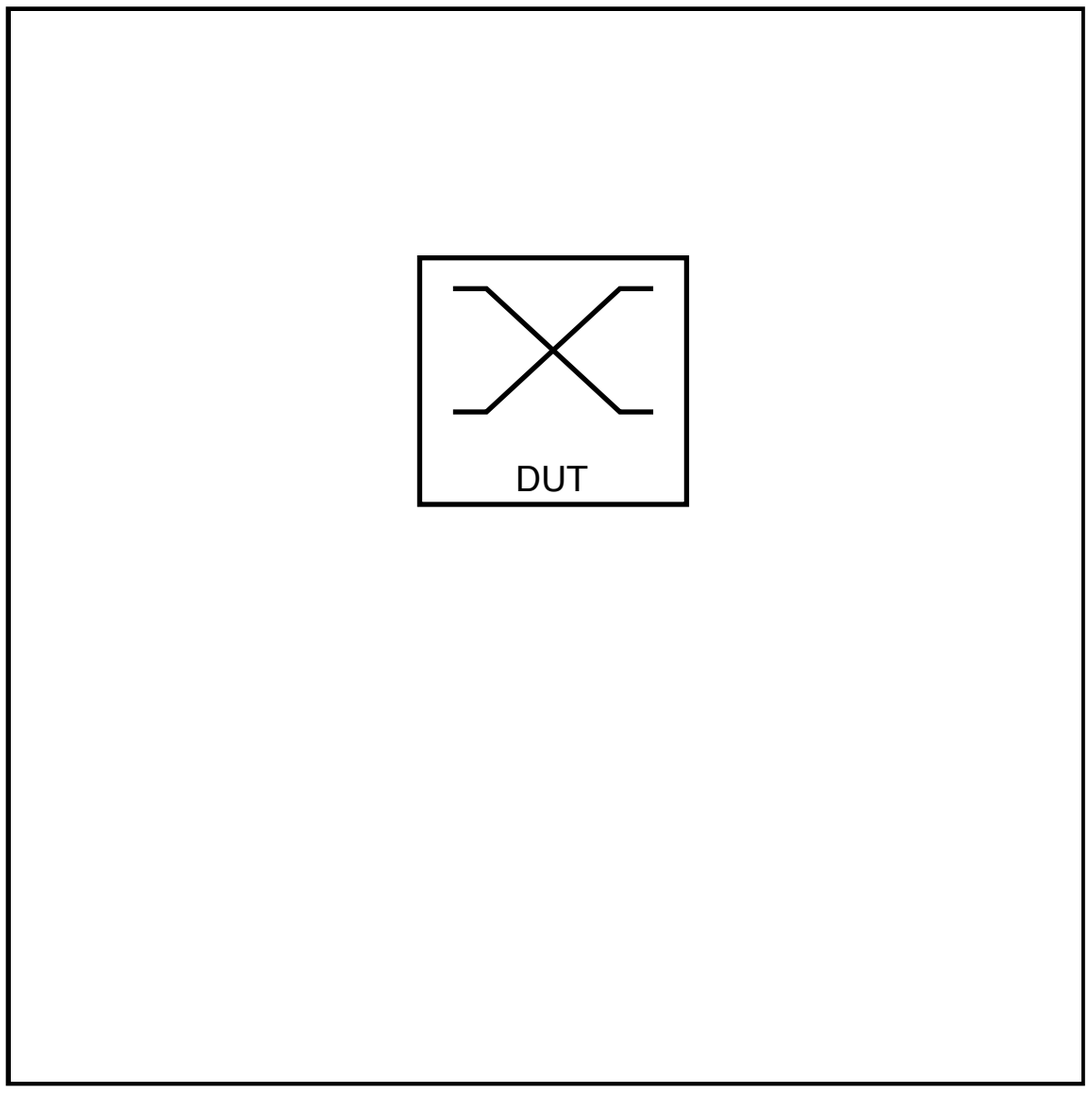
- One out of one million frames
  - Reduced data rate until all frames were captured
  - Note the more or less constant frame rate of 1.5Mfps
  - PCI-X bus speed harms larger frame sizes.
- One out of 16 frames
  - Lots of context switching
  - Again a more or less constant frame rate.

Generic purpose PC



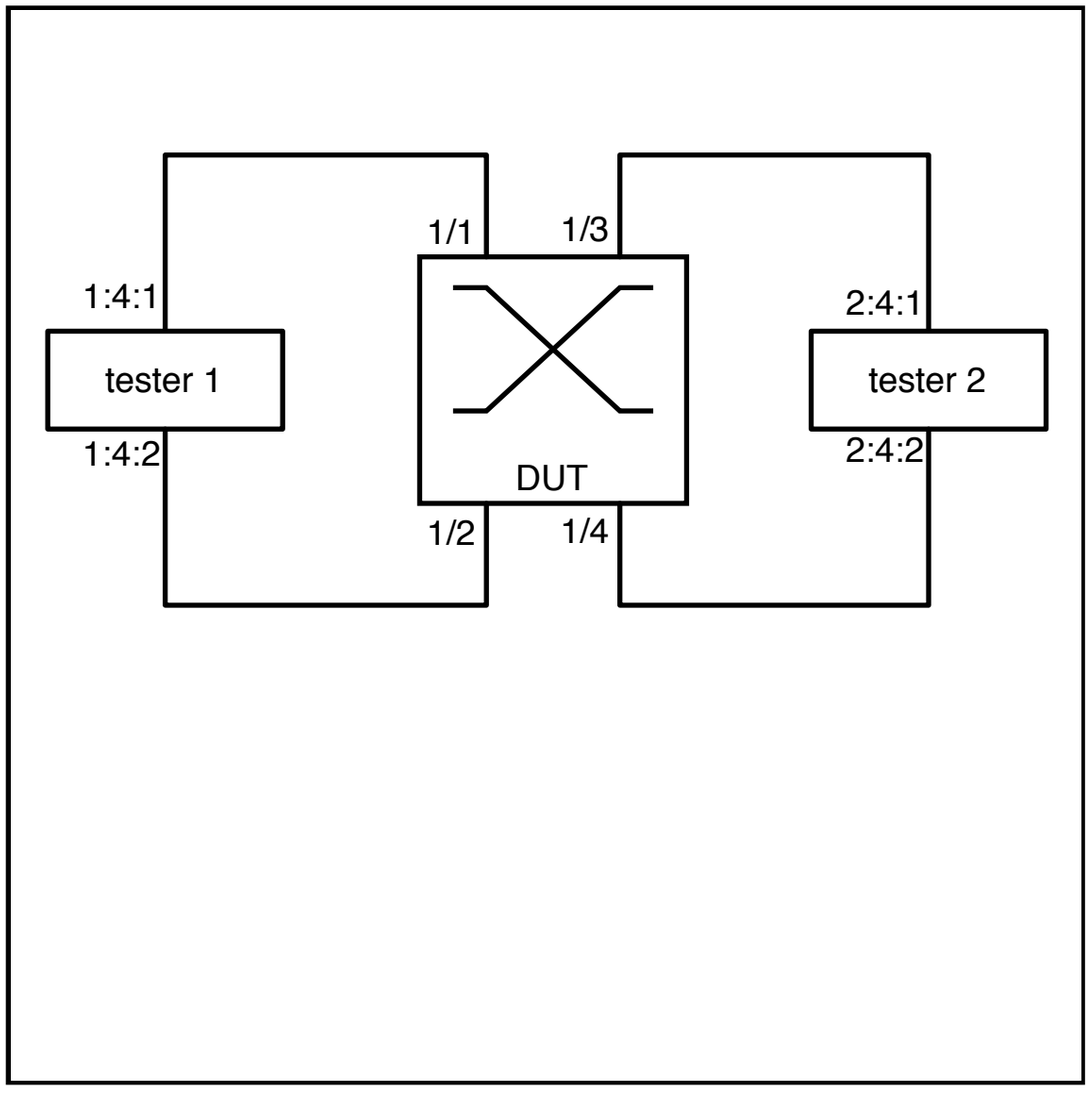
- One out of 16 frames
  - Lot's of context switching
  - Again a more or less constant frame rate.
- Conclusion
  - General purpose hardware is not fast enough.

Mirror Port



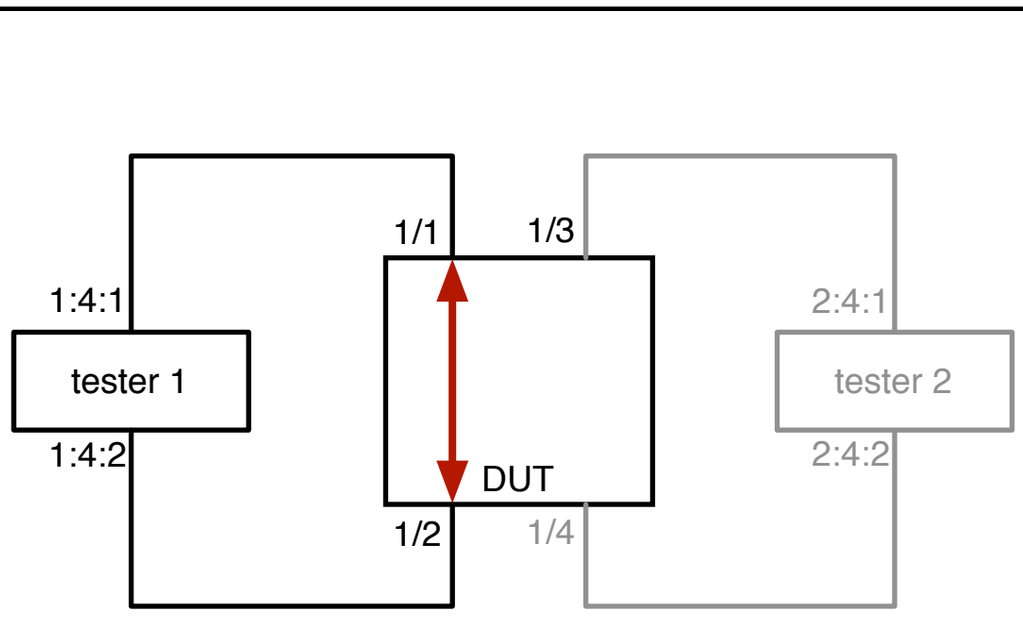
- Device Under Test (DUT)
  - Foundry Networks RX series

Mirror Port



- Device Under Test (DUT)
  - Foundry Networks RX series

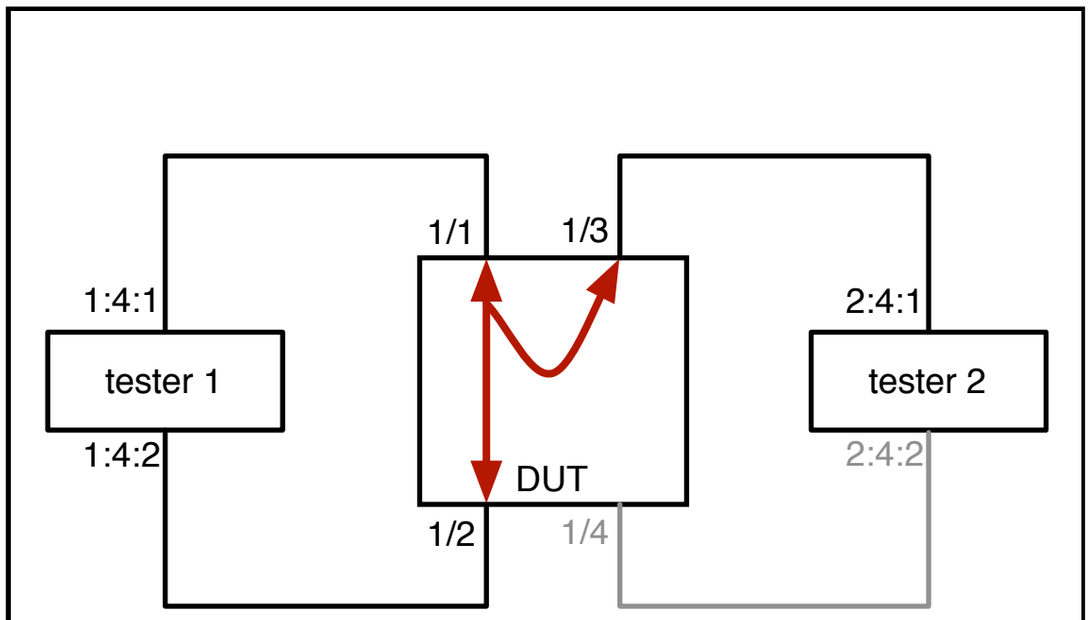
Mirror Port



frame size (bytes)	throughput (%)
64	100
128	100
256	100
512	100
1024	100
1280	100
1500	100

- Device Under Test (DUT)
  - Foundry Networks RX series
- Test 1
  - Throughput without monitoring.
  - 100% at all frame rates

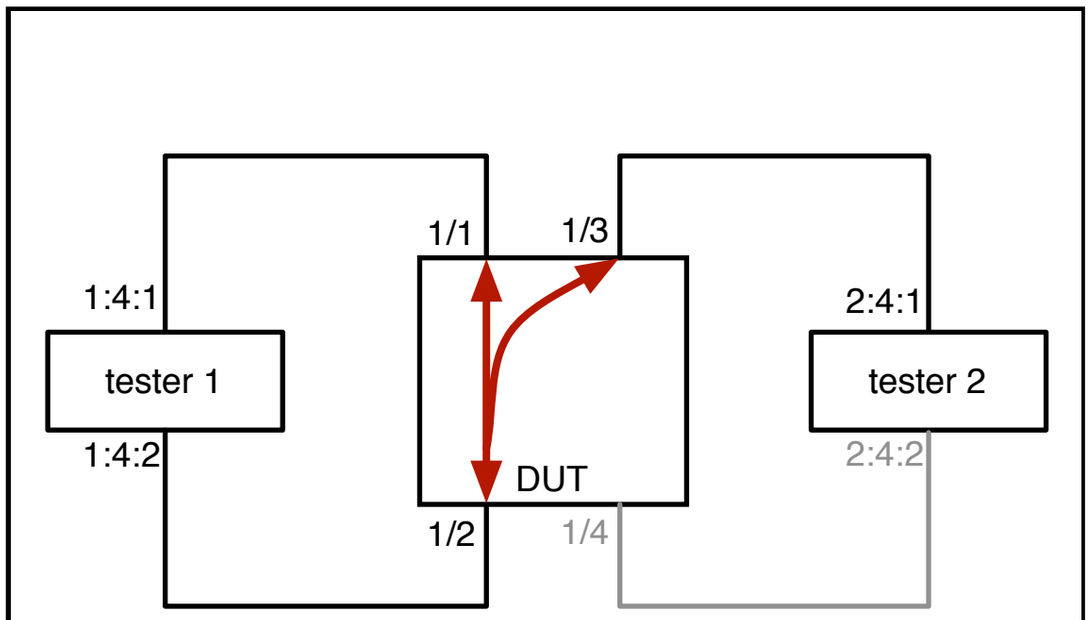
Mirror Port



frame size (bytes)	throughput (%)
64	56.95
128	58.78
256	70.00
512	68.66
1024	67.05
1280	69.26
1500	65.77

- Device Under Test (DUT)
  - Foundry Networks RX series
- Test 2
  - Ingress monitoring
  - Input traffic on port 1/1 is copied to port 1/3
- Test 3
  - Egress monitoring
  - Output traffic to port 1/1 is copied to 1/3.

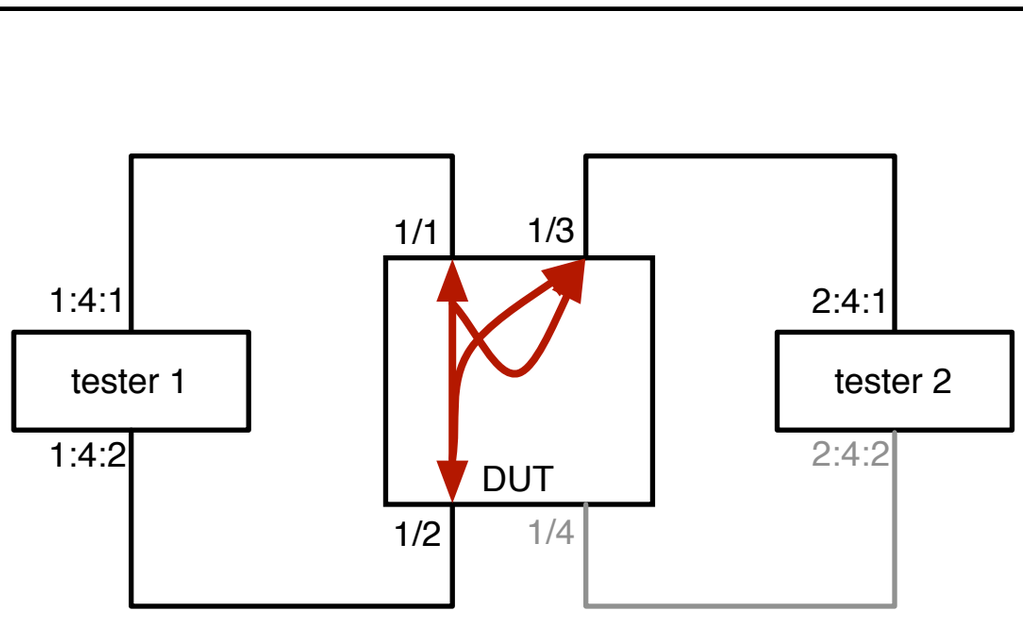
Mirror Port



frame size (bytes)	throughput (%)
64	56.95
128	58.76
256	70.84
512	68.66
1024	67.05
1280	69.27
1500	65.70

- Device Under Test (DUT)
  - Foundry Networks RX series
- Test 2
  - Ingress monitoring
  - Input traffic on port 1/1 is copied to port 1/3
- Test 3
  - Egress monitoring
  - Output traffic to port 1/1 is copied to 1/3.

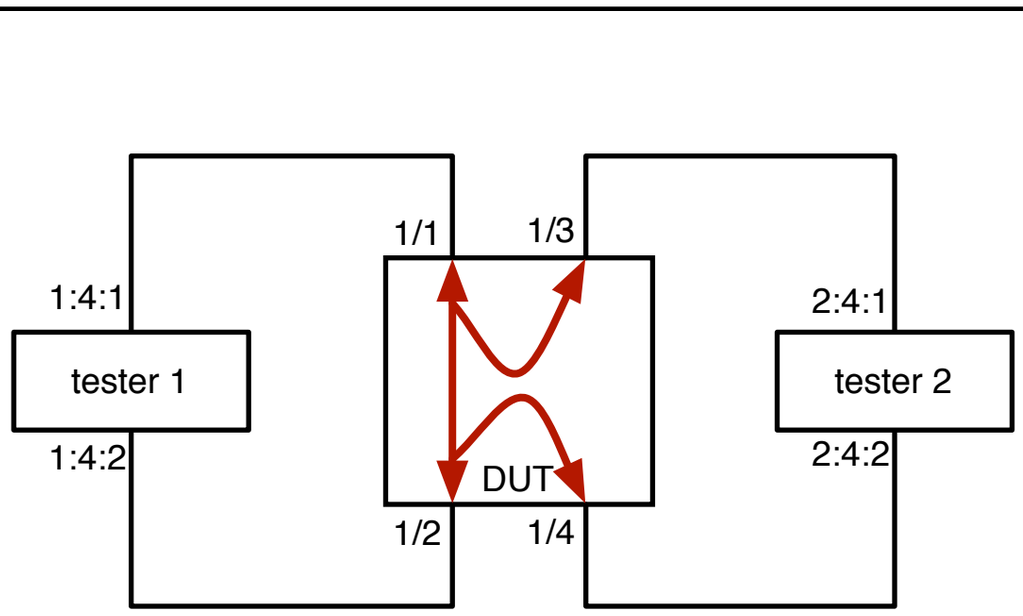
Mirror Port



frame size (bytes)	throughput (%)
64	45.78
128	43.23
256	50.15
512	46.88
1024	45.95
1280	51.36
1500	44.99

- Device Under Test (DUT)
  - Foundry Networks RX series
- Test 3
  - Egress monitoring:
  - Output traffic to port 1/1 is copied to 1/3.
- Test 4
  - Both ingress and egress monitoring to one mirror port:
  - Bi-directional traffic at port 1/1 is copied to port 1/3.

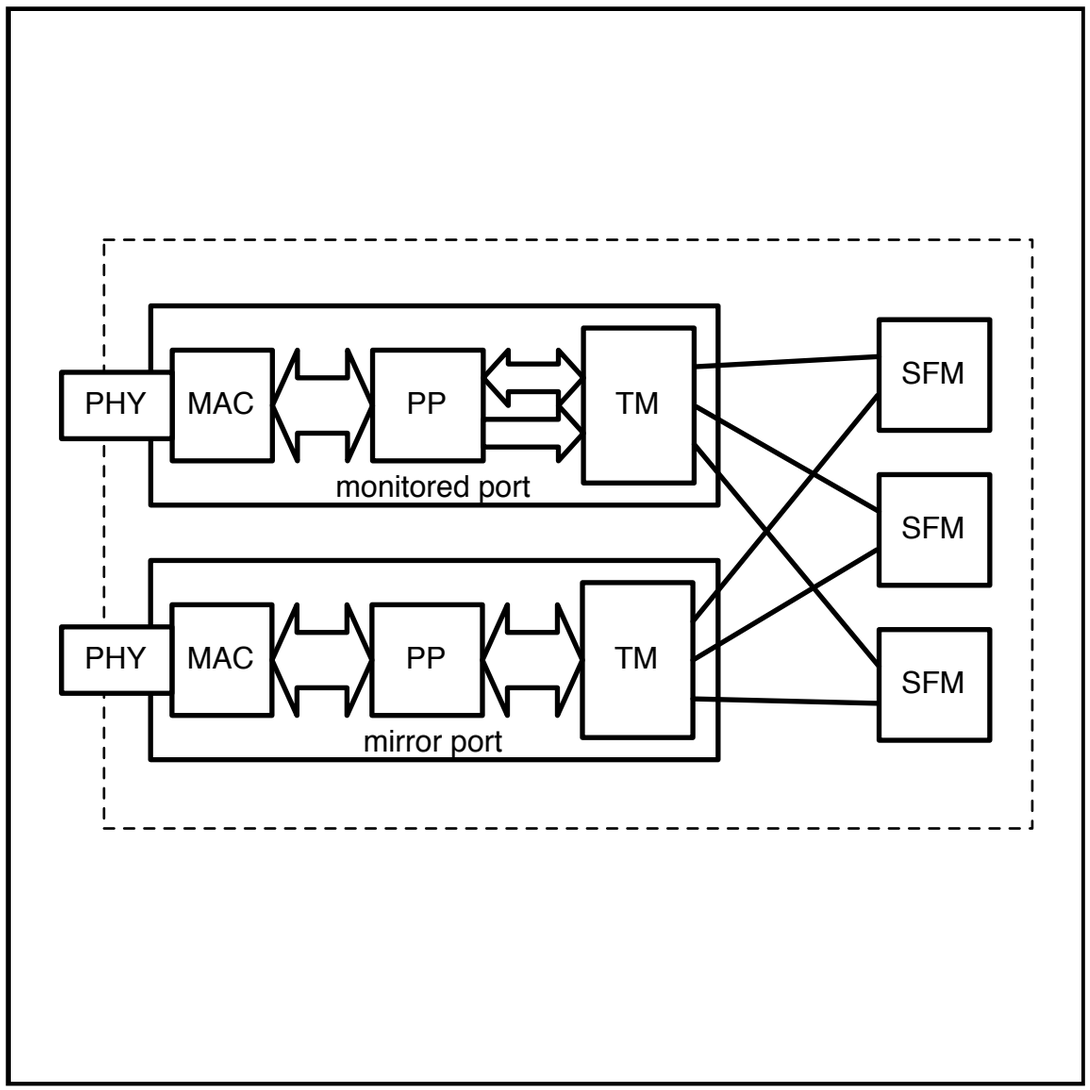
Mirror Port



frame size (bytes)	throughput (%)
64	37.39
128	38.35
256	44.82
512	45.69
1024	44.76
1280	46.63
1500	43.91

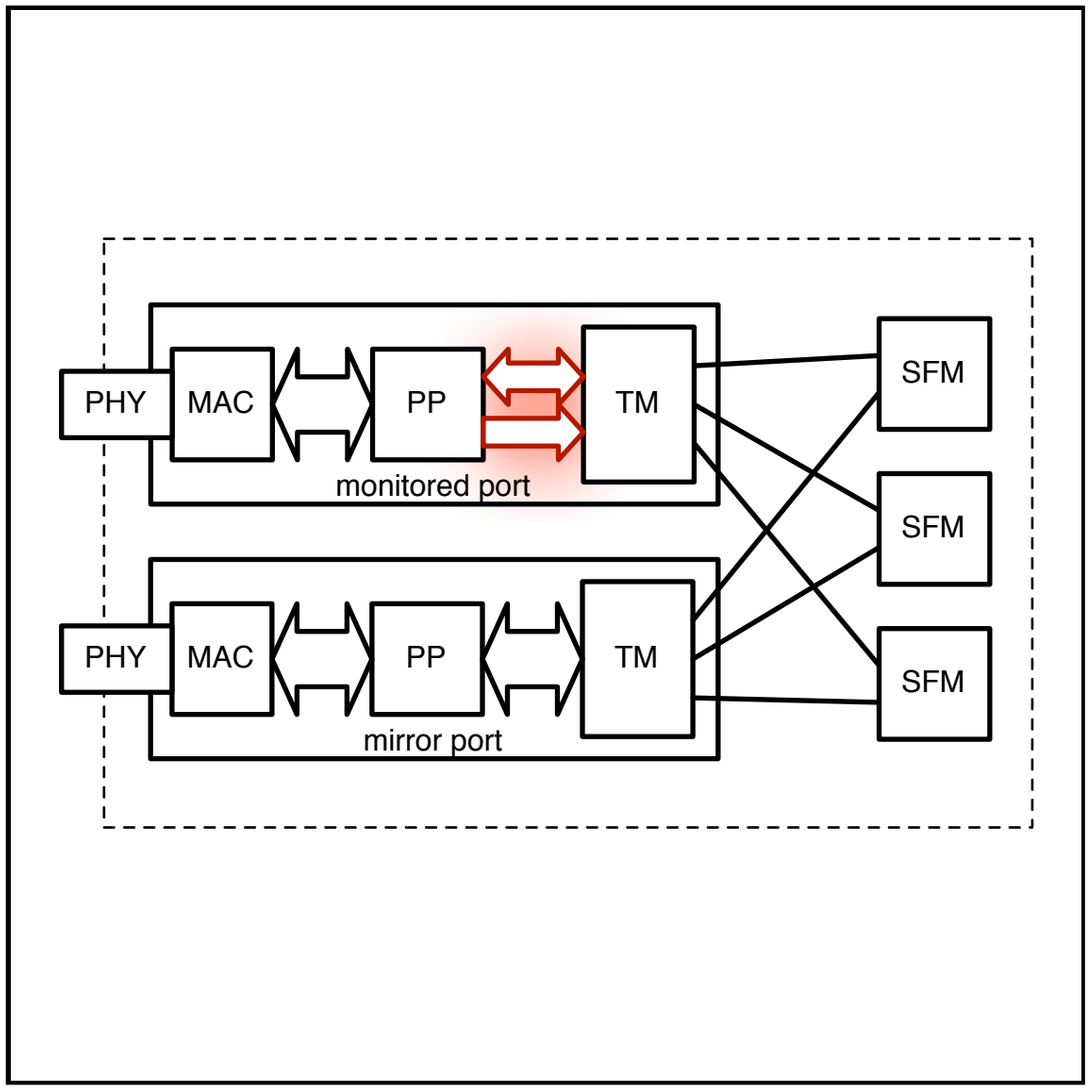
- Device Under Test (DUT)
  - Foundry Networks RX series
- Test 4
  - Both ingress and egress monitoring to one mirror port:
  - Bi-directional traffic at port 1/1 is copied to port 1/3.
- Test 5
  - Ingress traffic at port 1/1 is copied to port 1/3
  - Output traffic to port 1/1 is copied to 1/4.

Mirror Port



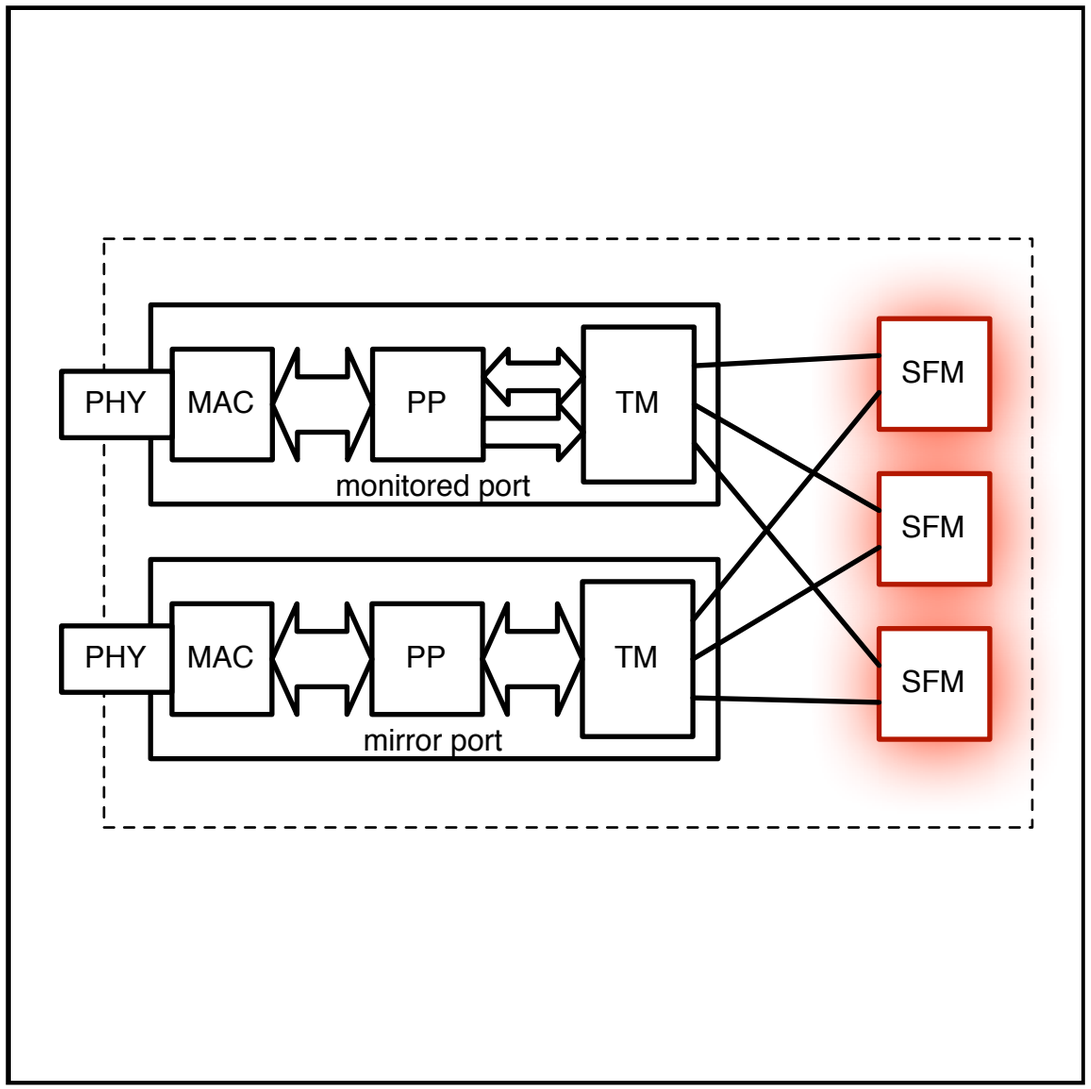
- Device Under Test (DUT)
  - Foundry Networks RX series
- Conclusion
  - No mirroring: no frame loss
  - Mirroring: frame loss above certain data rates.
    - Best case: 70% throughput
    - Worst case: 36% throughput

Mirror Port



- Device Under Test (DUT)
  - Foundry Networks RX series
- Conclusion
  - No mirroring: no frame loss
  - Mirroring: frame loss above certain data rates.
    - Best case: 70% throughput
    - Worst case: 36% throughput
- Result of design decisions

Mirror Port

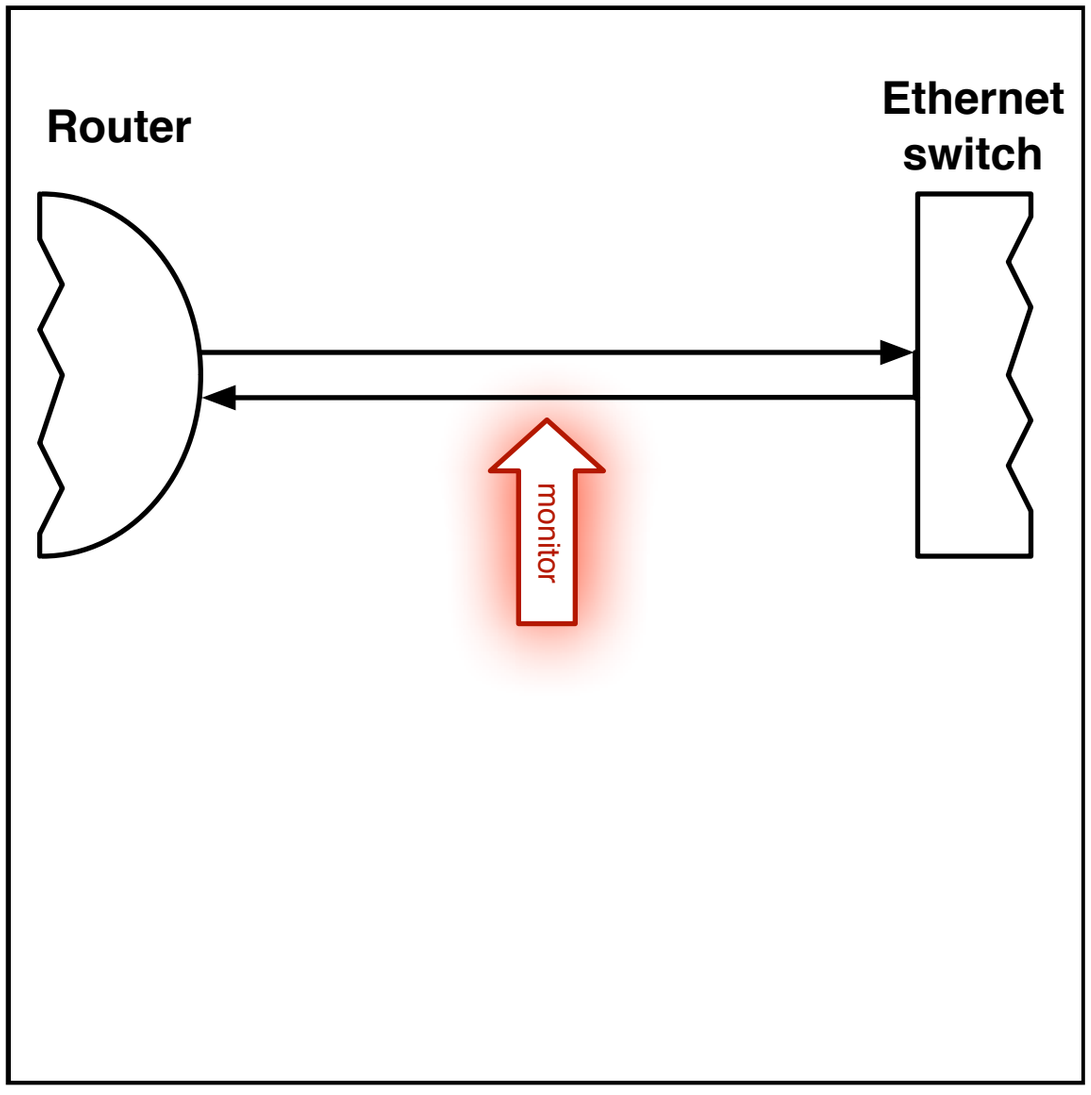


- Device Under Test (DUT)
  - Foundry Networks RX series
- Conclusion
  - No mirroring: no frame loss
  - Mirroring: frame loss above certain data rates.
    - Best case: 70% throughput
    - Worst case: 36% throughput
- Result of design decisions
  - Replication takes place in packet processor.

Agenda

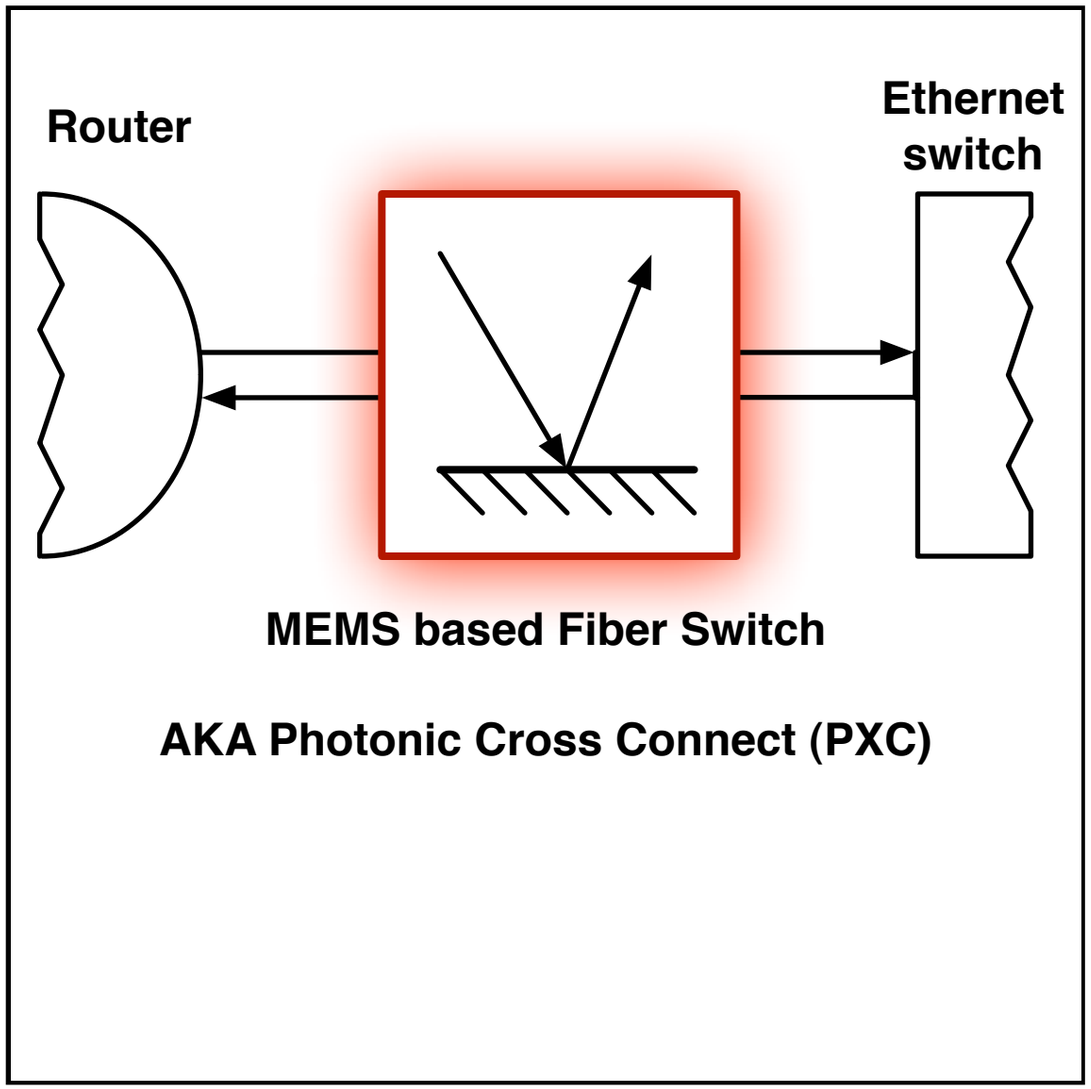
- An example.
- The problems with passive 10GbE monitoring.
  - Performance of generic computer hardware.
  - Issues with the mirror port feature.
- Overview of our solution.
  - Hardware architecture.
  - Homegrown firmware and software.
- A bit more detail.
- More to demo.

Solution



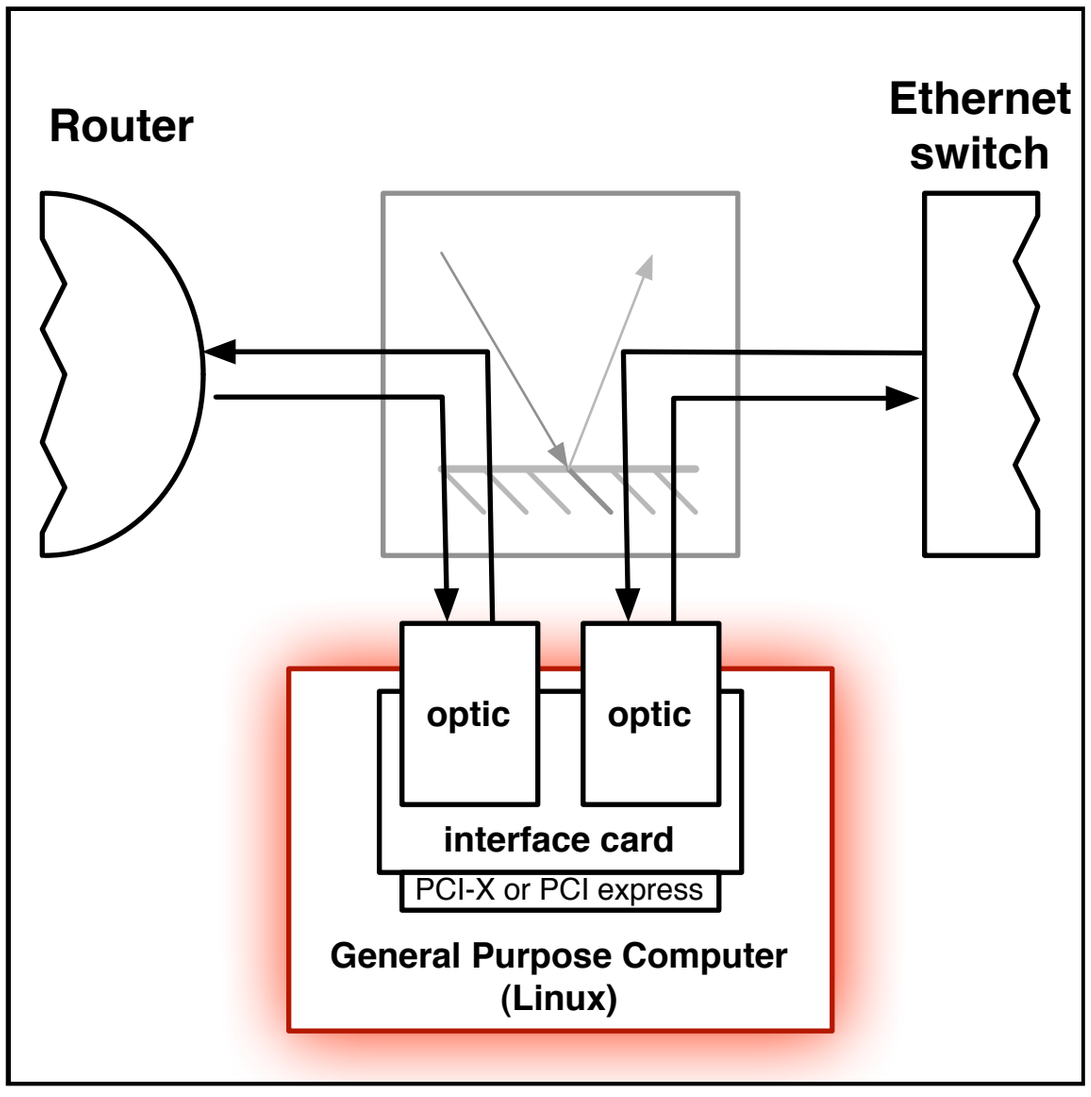
- We need to monitor in-line.
  - We can use our PXCs...

Solution



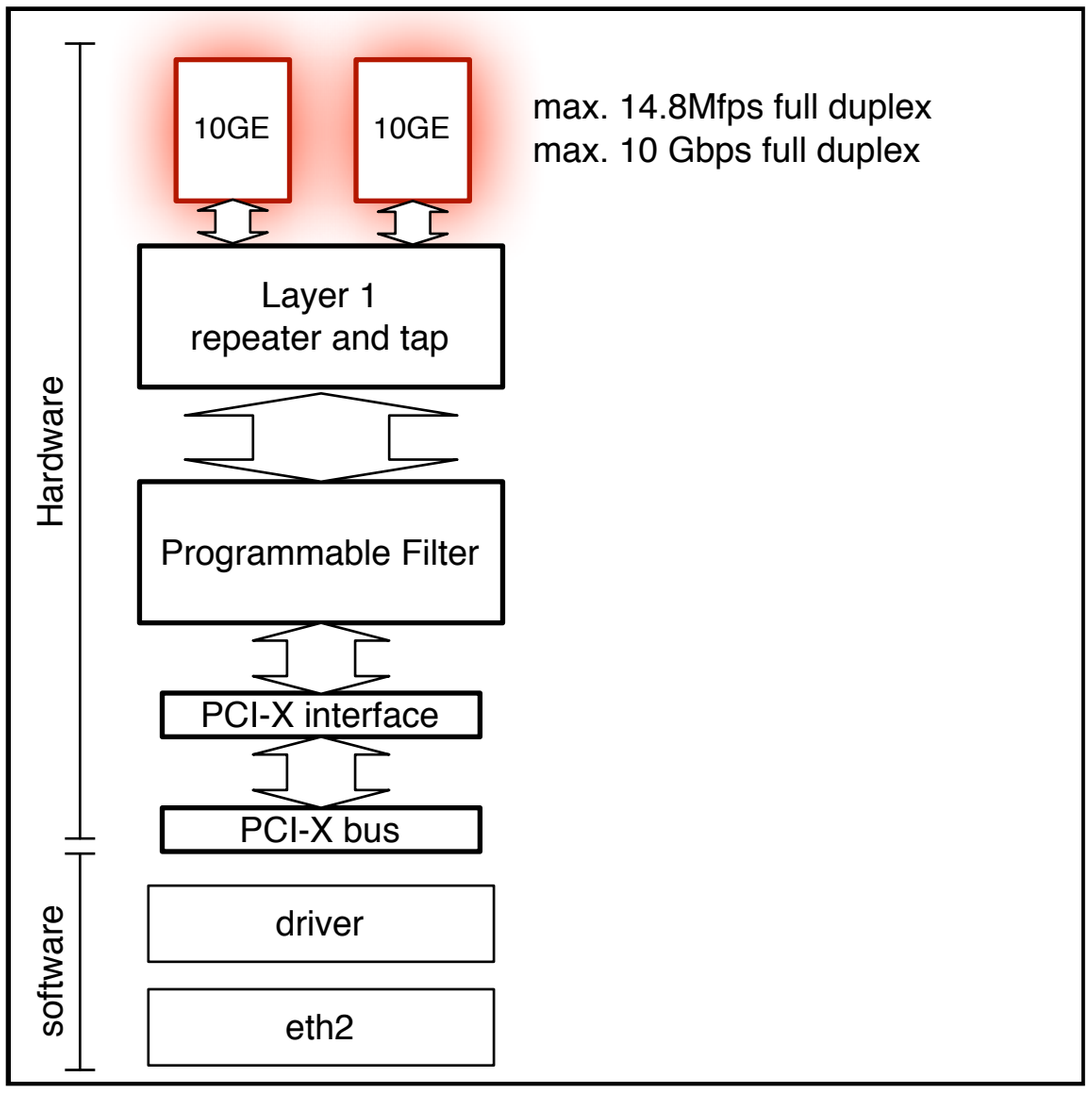
- We need to monitor in-line.
  - We can use our PXCs...

Solution



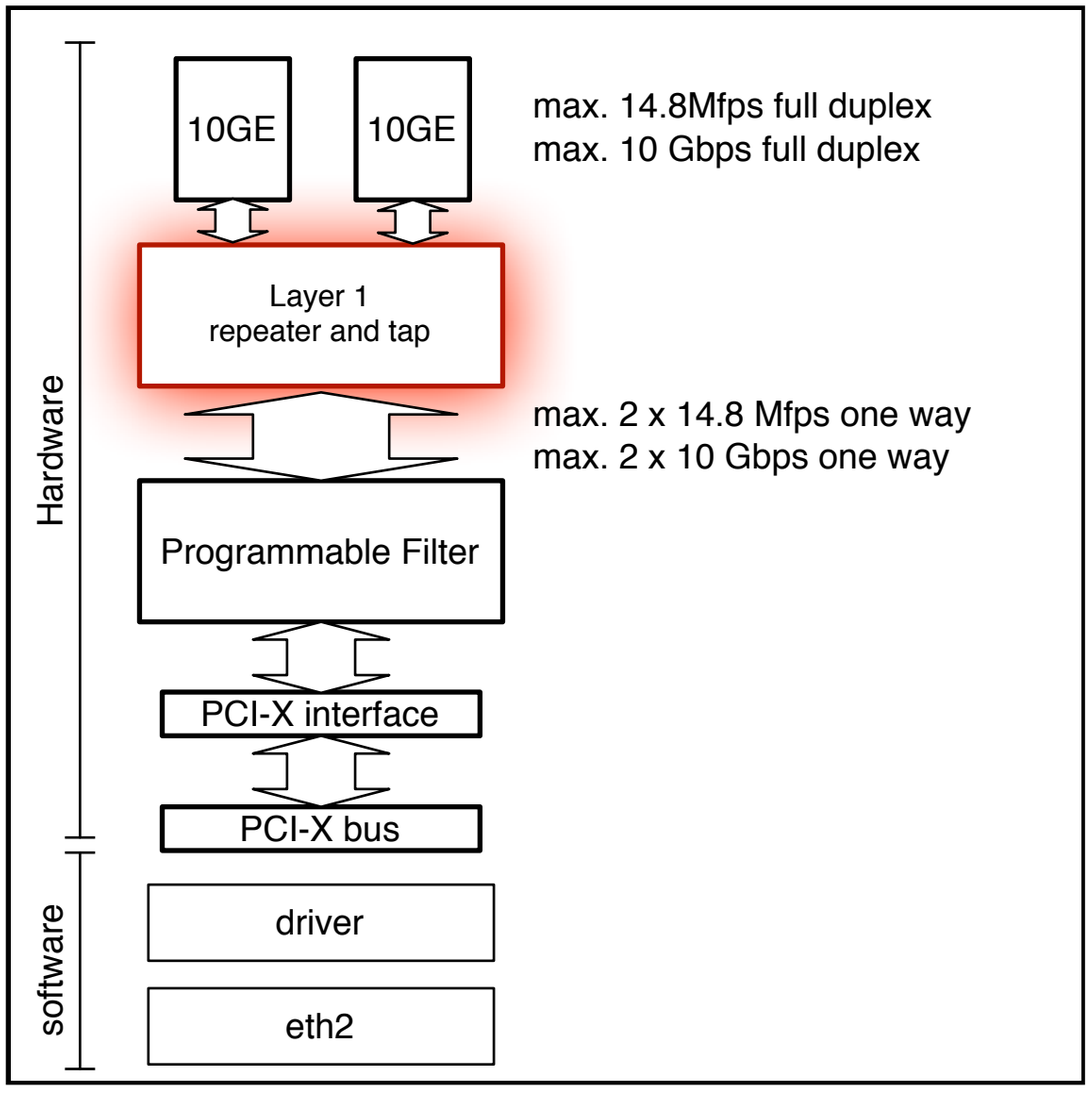
- We need to monitor in-line.
  - We can use our PXCs... to insert some kind of tap or repeater.

System Overview



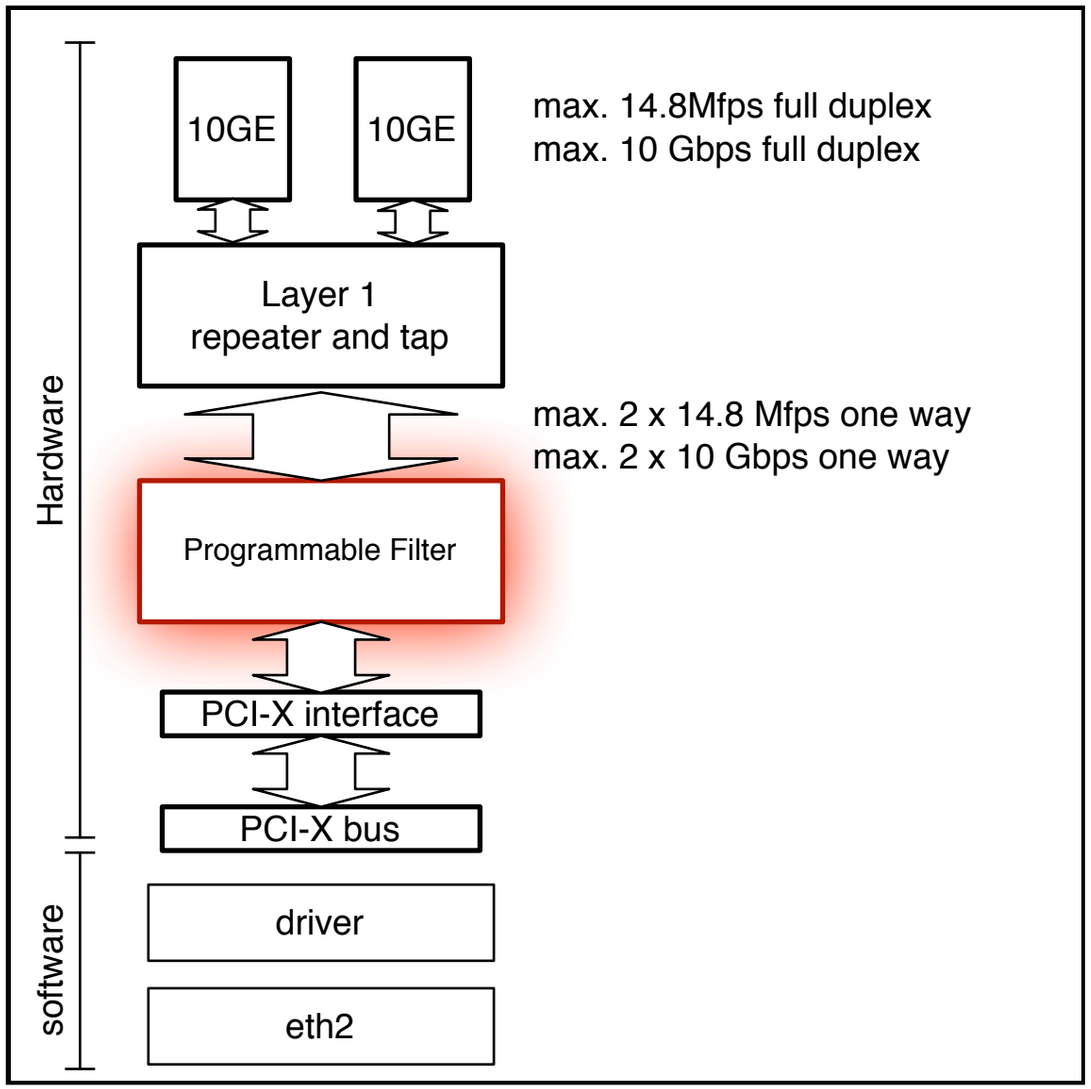
- We need to monitor in-line.
  - We can use our PXCs... to insert some kind of tap or repeater.
- Two full duplex transceivers
  - In-line, for signal regeneration

System Overview



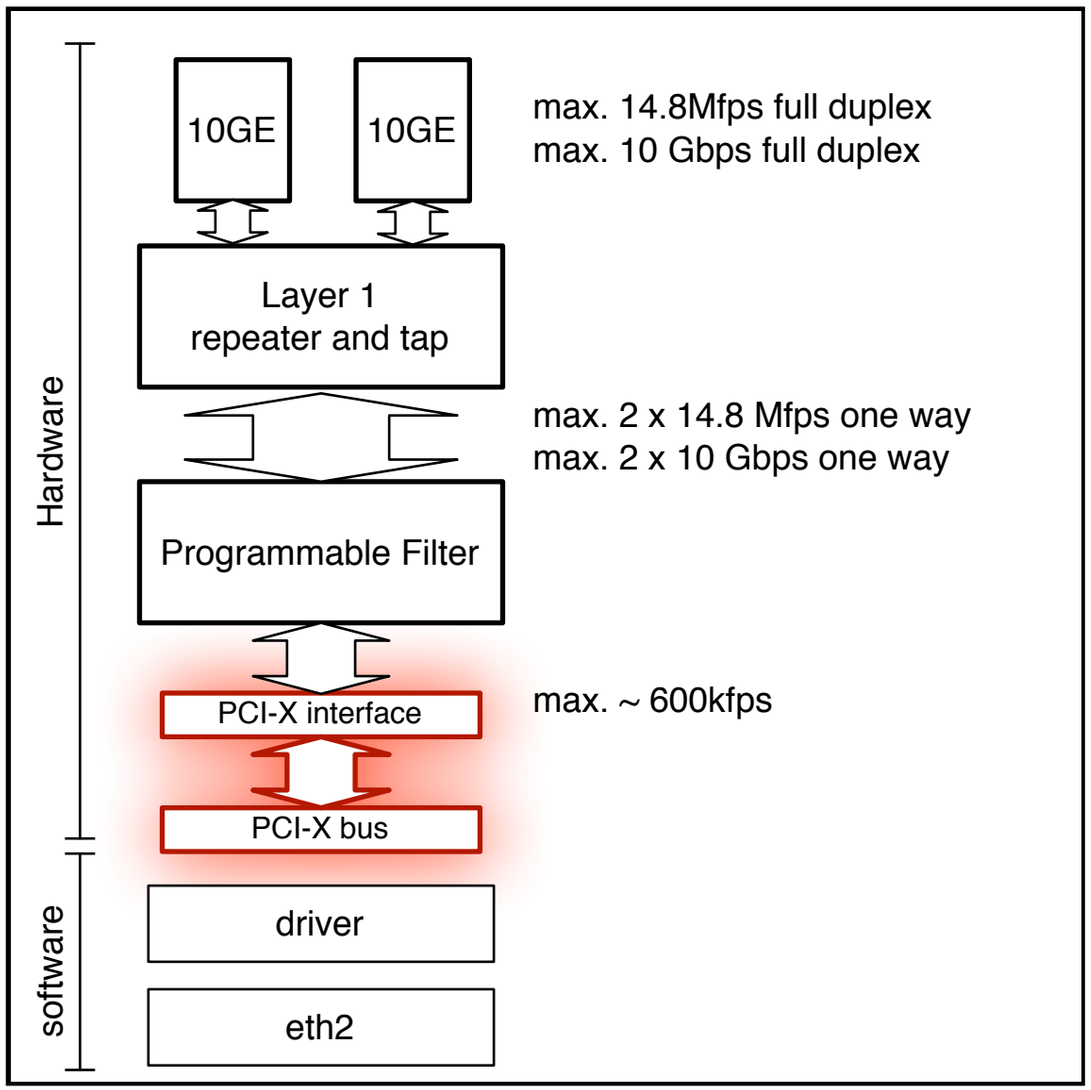
- We need to monitor in-line.
  - We can use our PXCs... to insert some kind of tap or repeater.
- Two full duplex transceivers
  - In-line, for signal regeneration
- Repeater and tap
  - Forwards traffic between transceivers and copies it for monitoring.

System Overview



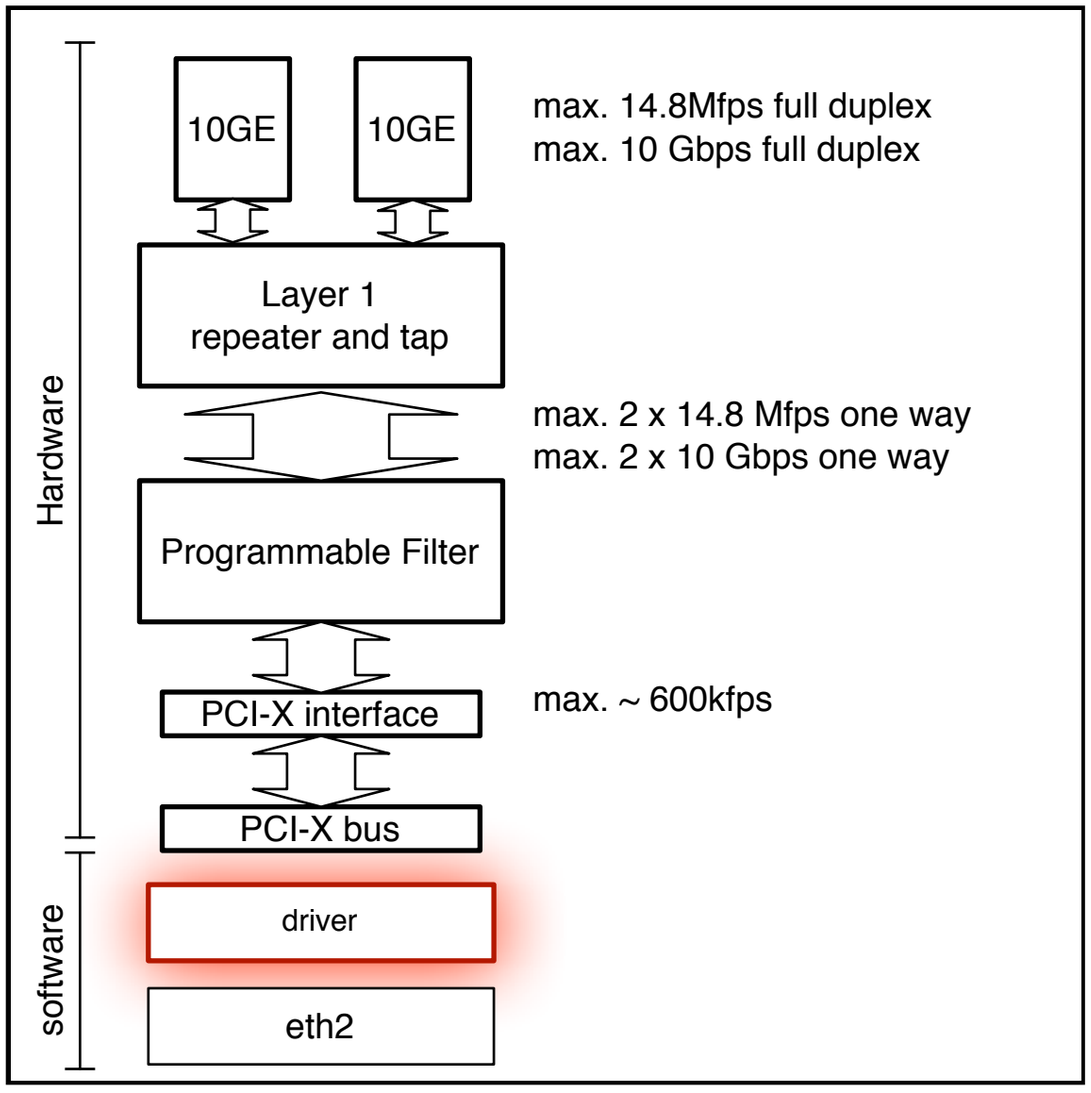
- Two full duplex transceivers
  - In-line, for signal regeneration
- Repeater and tap
  - Forwards traffic between transceivers and copies it for monitoring.
- User adjustable filter
  - Data rate reduction

System Overview



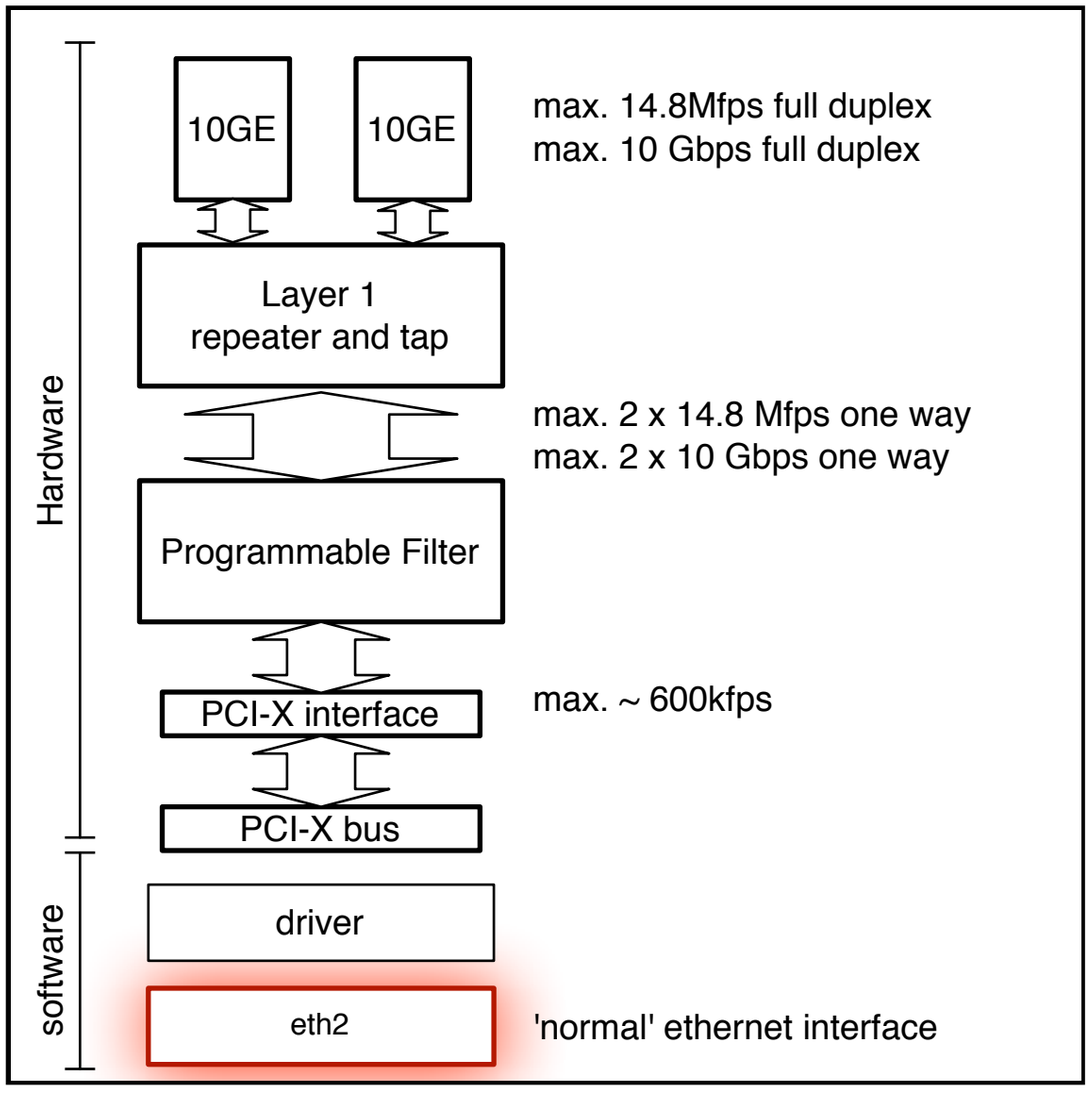
- Two full duplex transceivers
- Repeater and tap
  - Forwards traffic between transceivers and copies it for monitoring.
- User adjustable filter
  - Data rate reduction
- Standard interface bus
  - PCI-X bus

System Overview



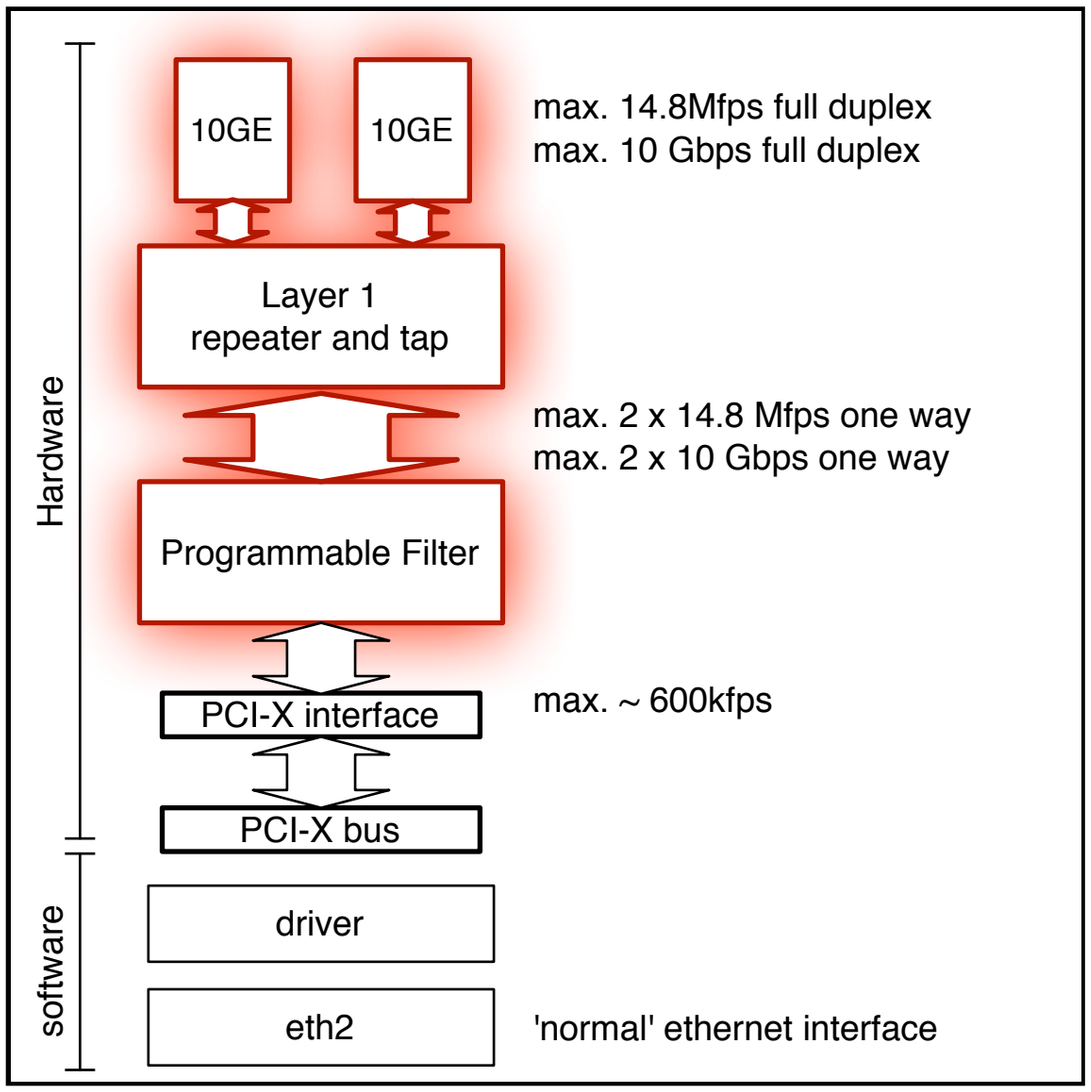
- Two full duplex transceivers
- Repeater and tap
  - Forwards traffic between transceivers and copies it for monitoring.
- User adjustable filter
  - Data rate reduction
- Standard interface bus
  - PCI-X bus
- Software
  - Kernel module to make it look like a ‘normal’ ethernet interface.

System Overview



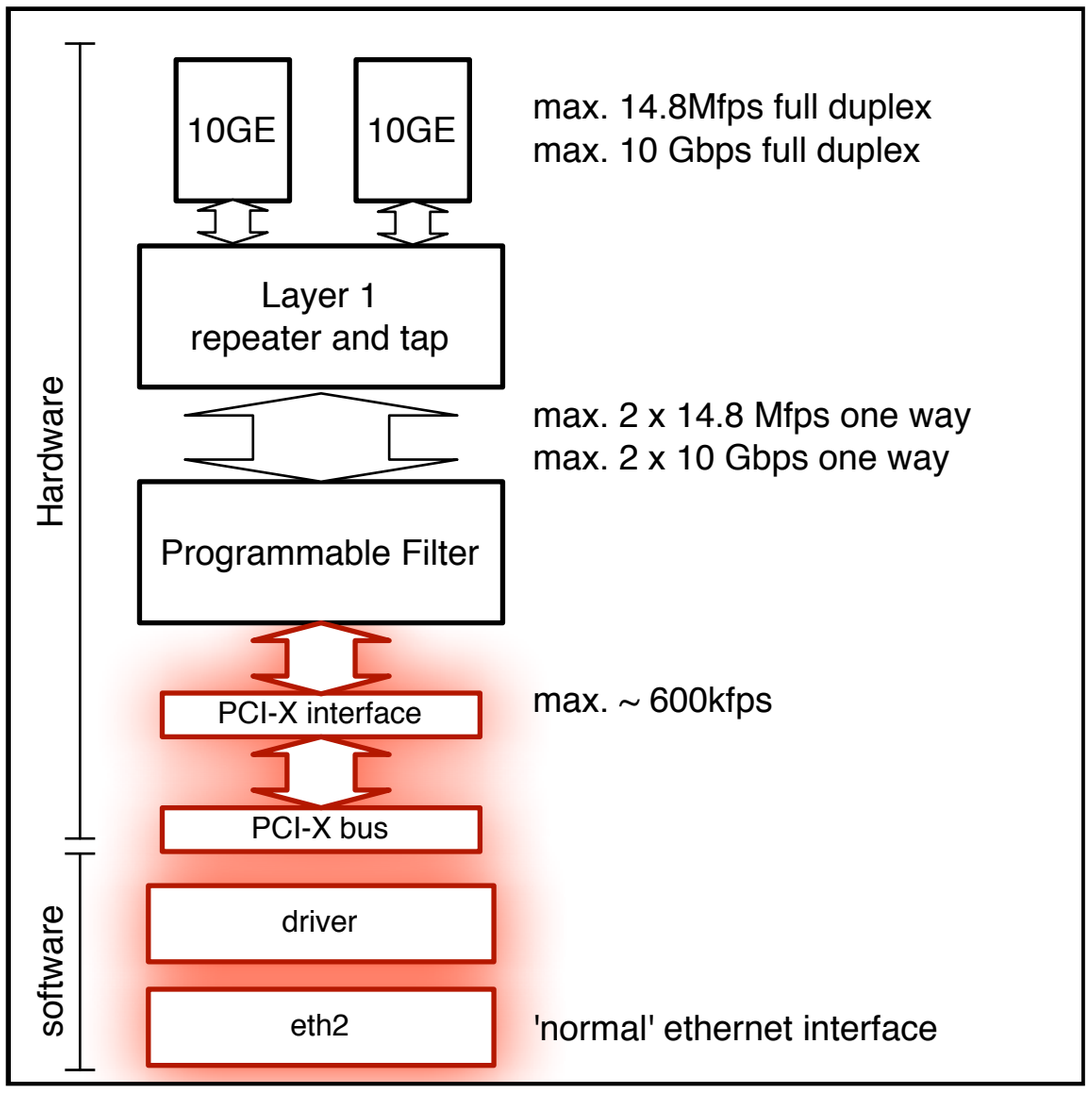
- Two full duplex transceivers
- Repeater and tap
  - Forwards traffic between transceivers and copies it for monitoring.
- User adjustable filter
  - Data rate reduction
- Standard interface bus
  - PCI-X bus
- Software
  - Kernel module to make it ‘look’ like a normal ethernet interface.

System Overview



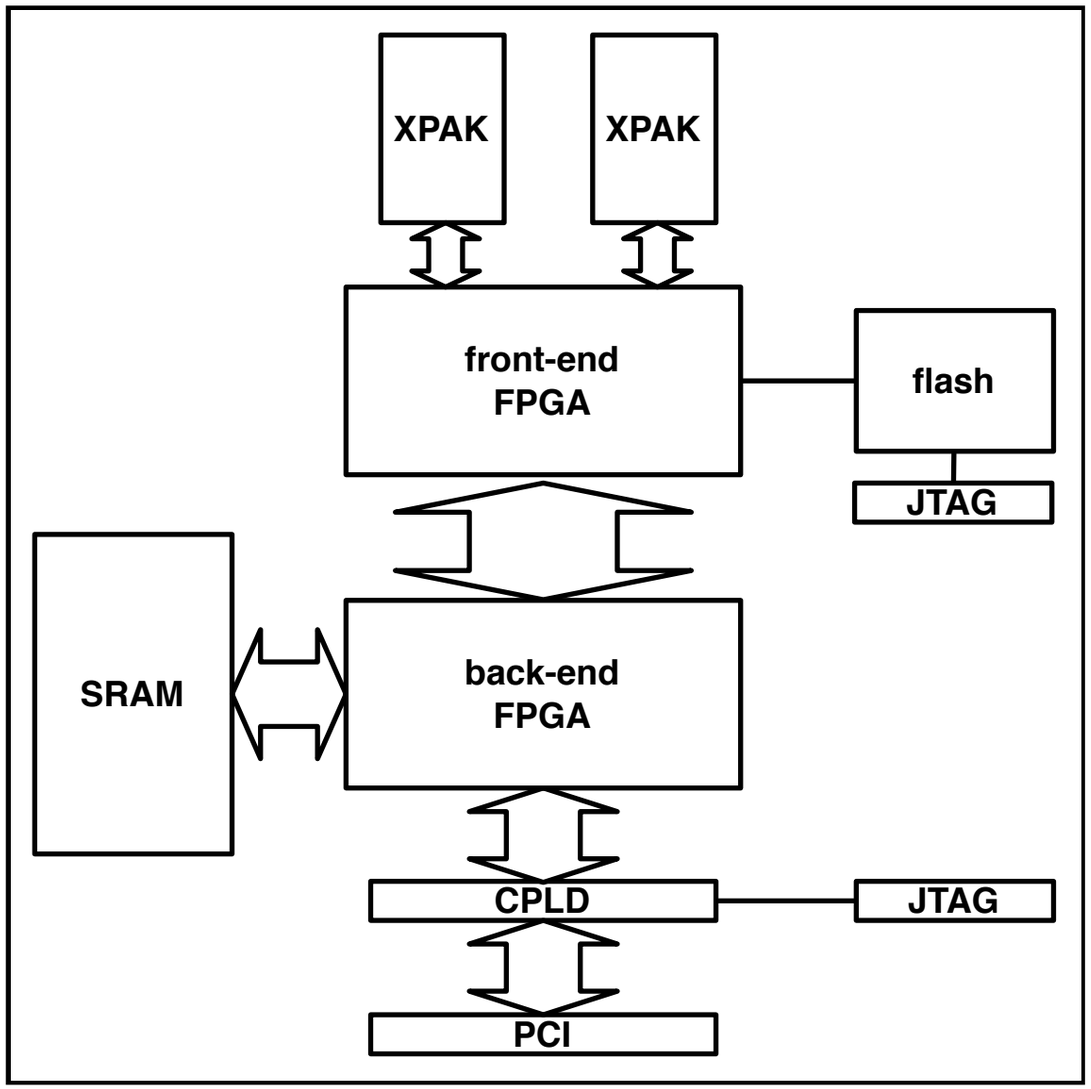
- Summarizing
  - Full line rate hardware

System Overview



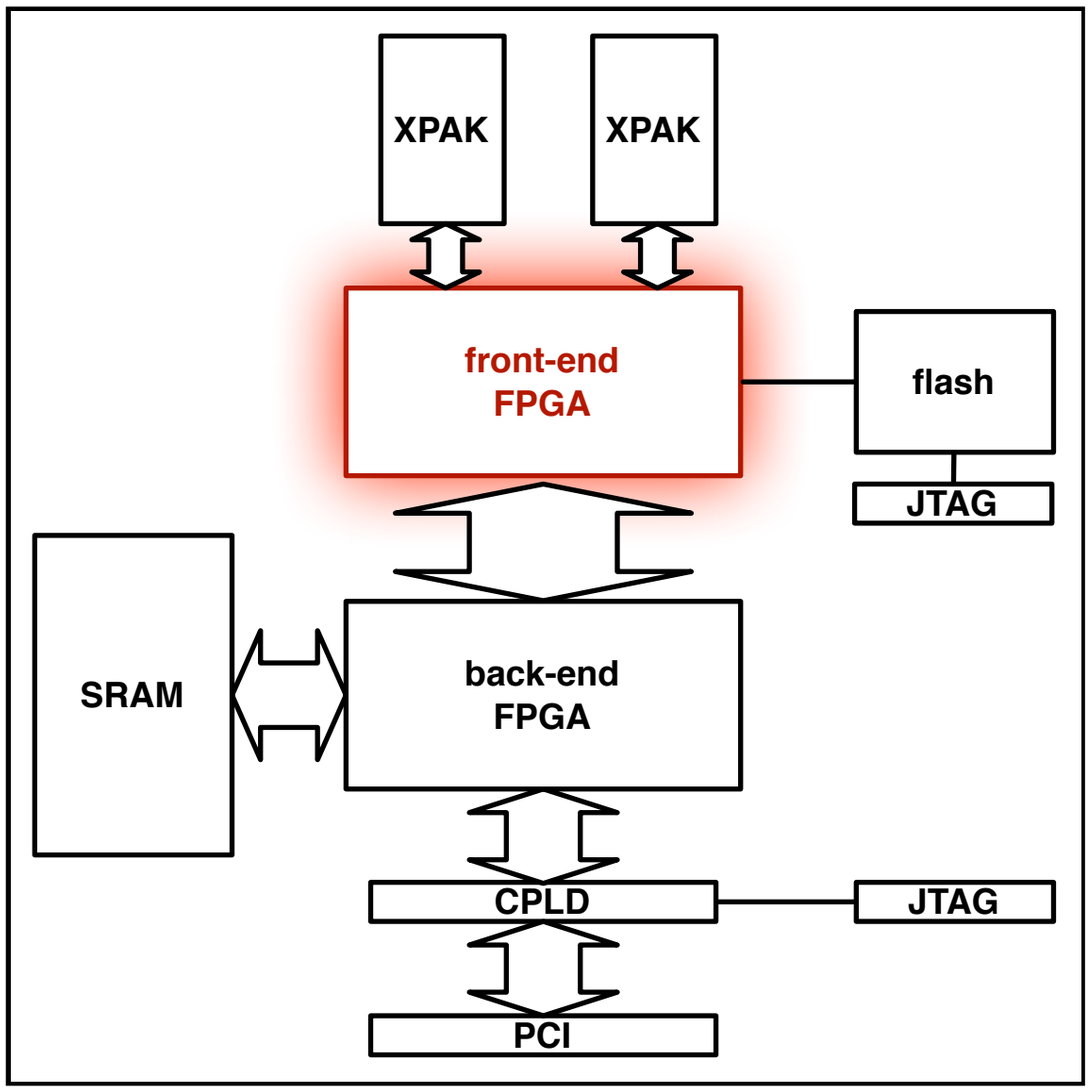
- Summarizing
  - Full line rate hardware
  - Lower the data rate
    so that the PC can handle
    it by either filtering or sampling

System Overview



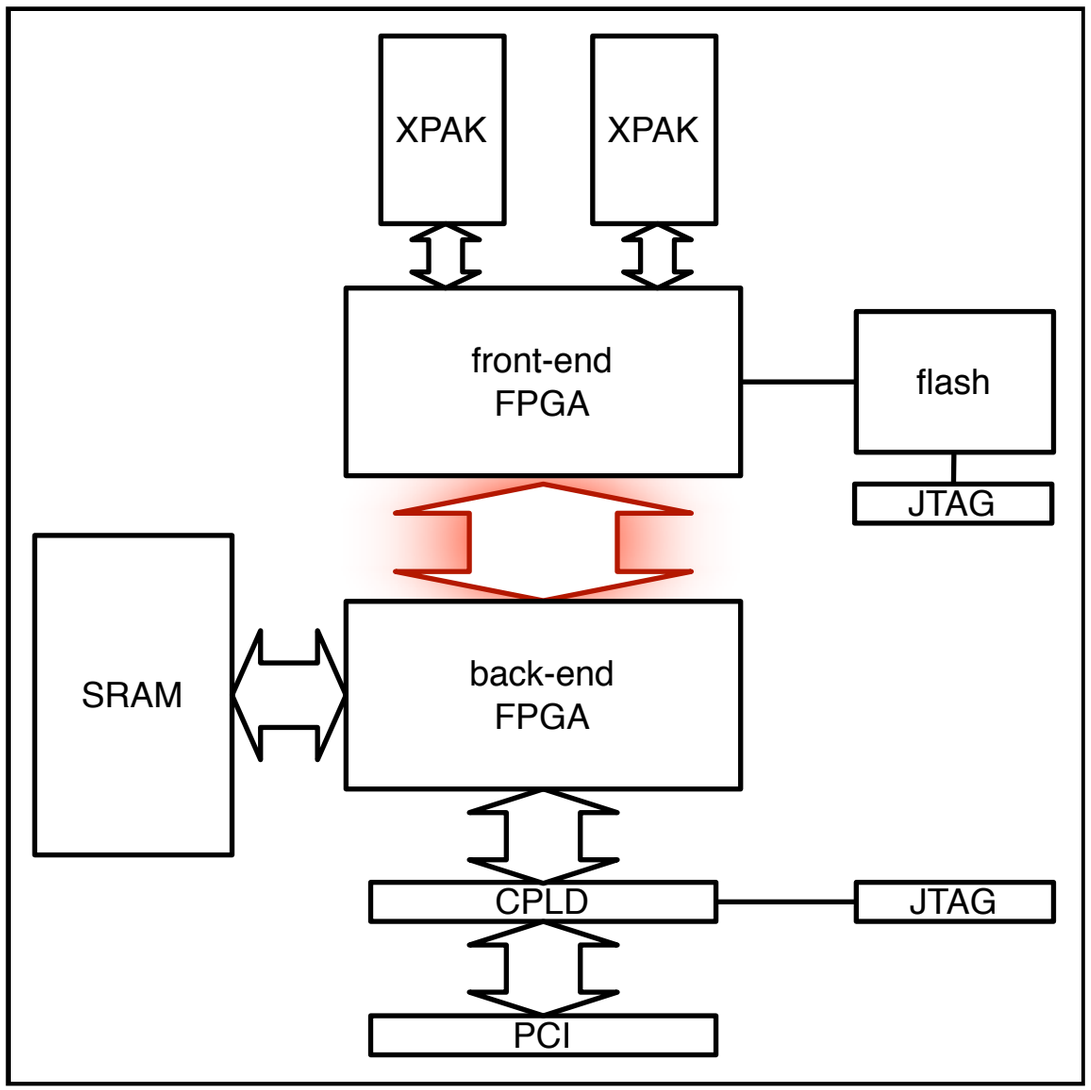
- Force10 MTP-10G
  - Network Interface Card NIC
  - Part of P10 IDS/IPS system
  - We wrote our own firmware and software for dynamic monitoring

System Overview



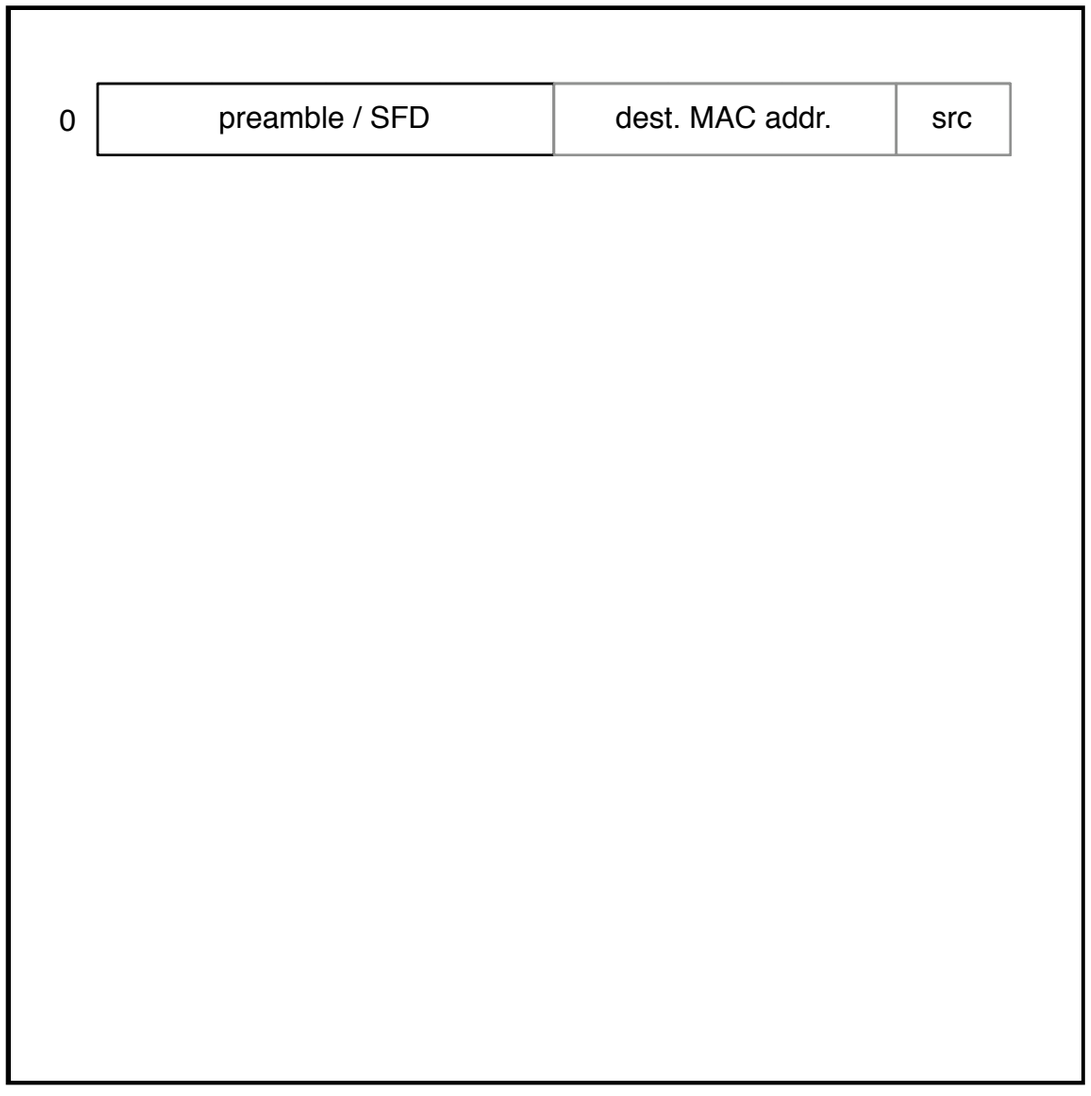
- Force10 MTP-10G
  - Network Interface Card NIC
  - Part of P10 IDS/IPS system
  - We wrote our own firm and software for dynamic monitoring
- Front-end FPGA
  - Full line rate tap and repeater

System Overview



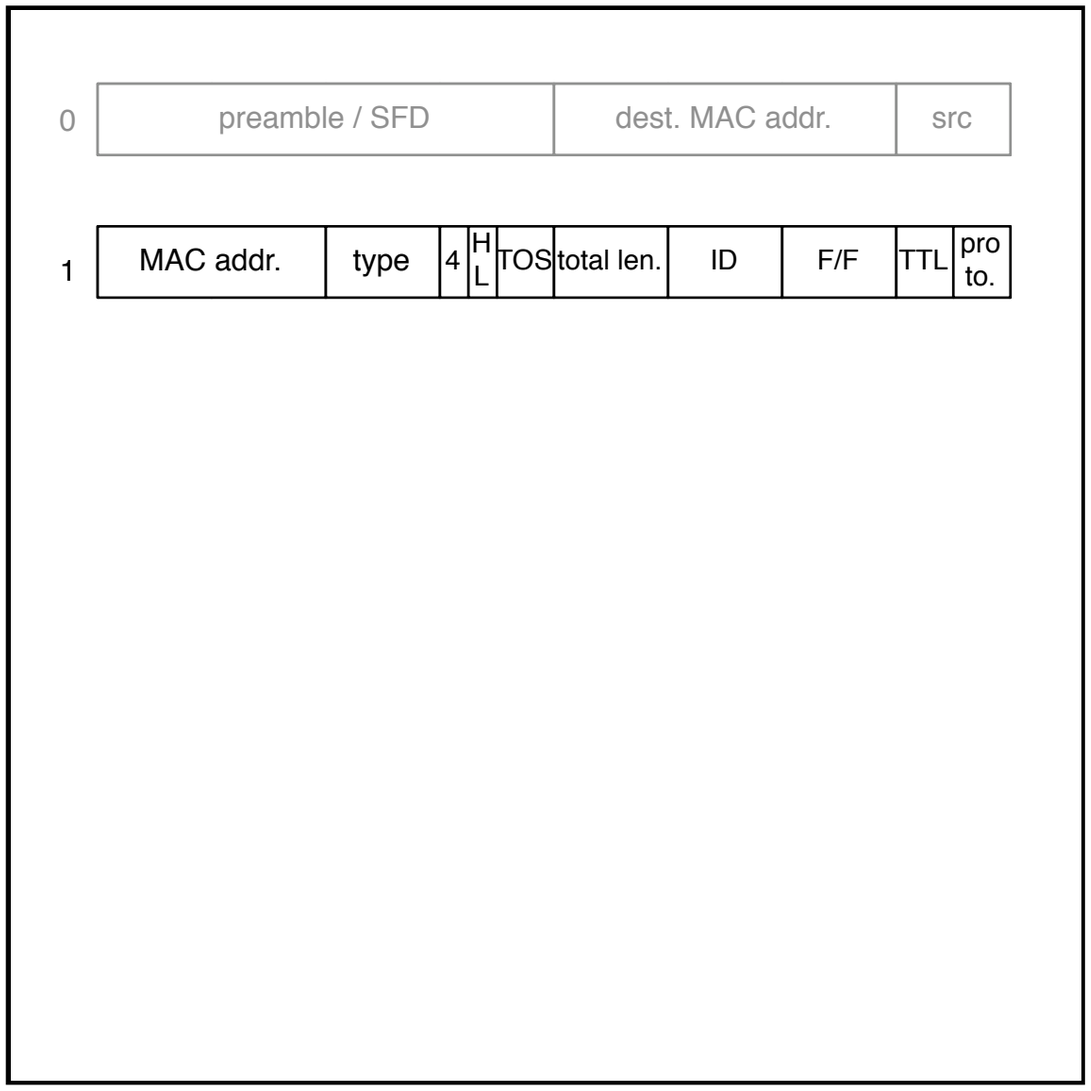
- Force10 MTP-10G
  - Network Interface Card NIC
  - Part of P10 IDS/IPS system
  - We wrote our own firm and software for dynamic monitoring
- Front-end FPGA
  - Full line rate tap and repeater
- Tapped traffic
  - 128-bit wide words

System Overview



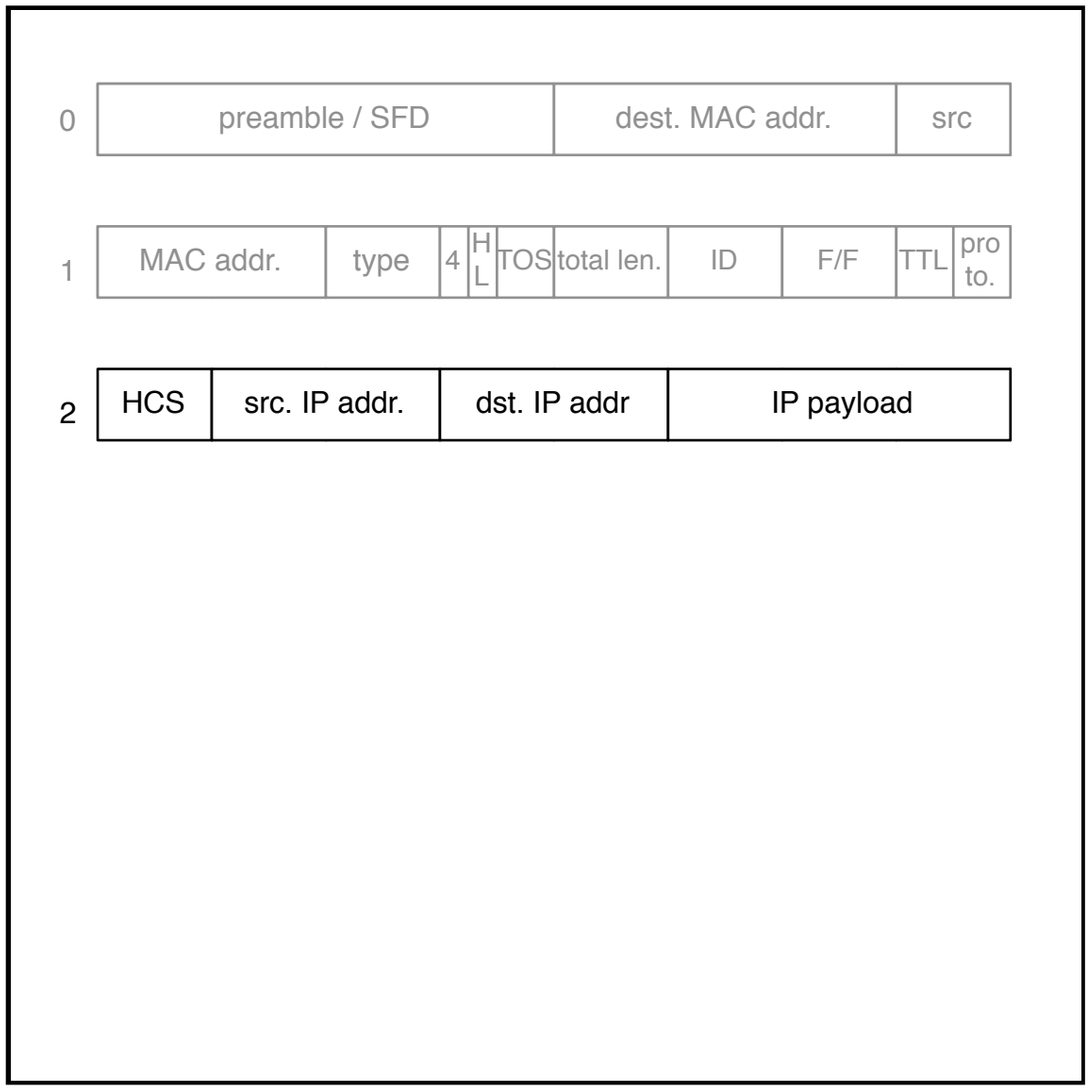
- Force10 MTP-10G
  - Network Interface Card NIC
  - Part of P10 IDS/IPS system
  - We wrote our own firm and software for dynamic monitoring
- Front-end FPGA
  - Full line rate tap and repeater
- Tapped traffic
  - 128-bit wide words

System Overview



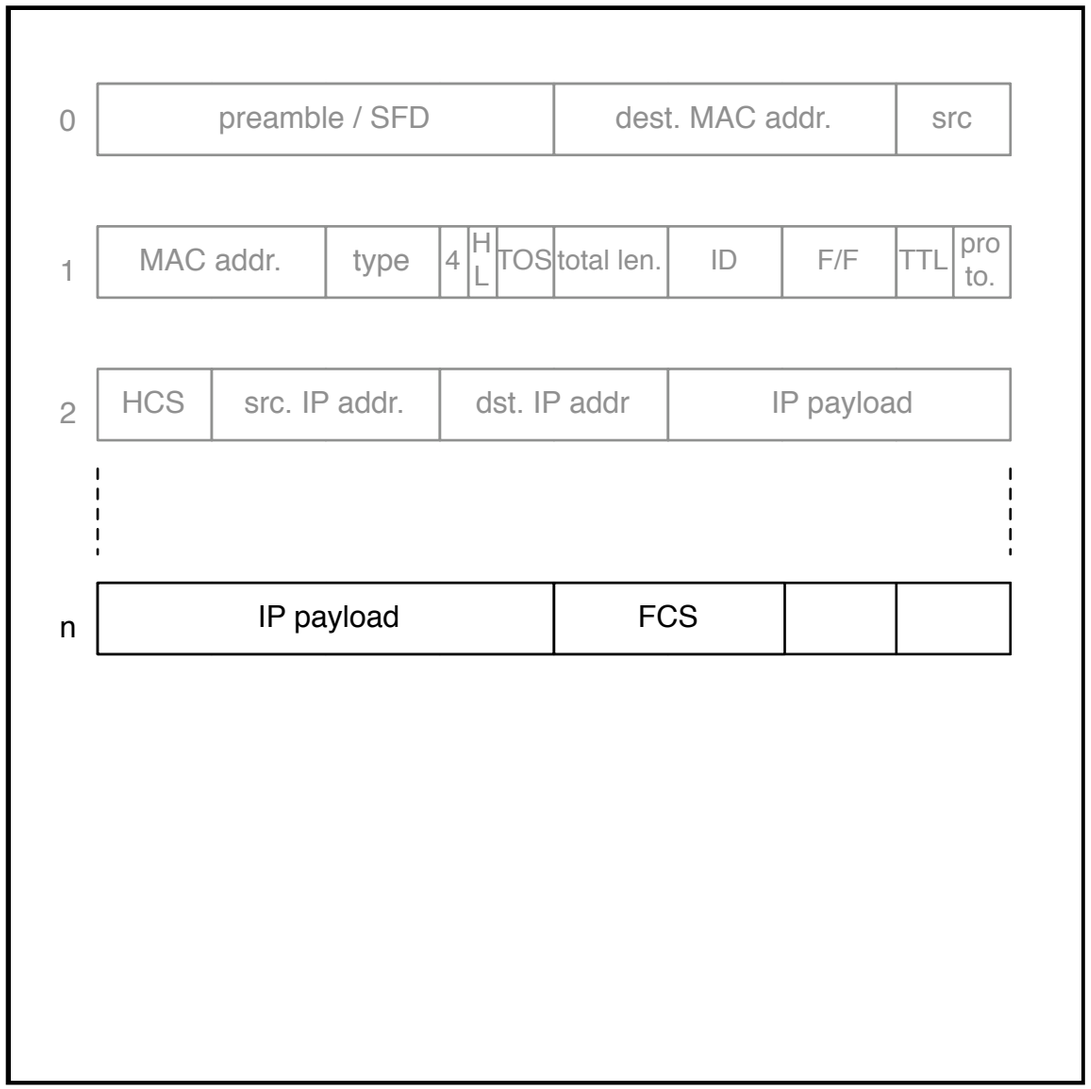
- Force10 MTP-10G
  - Network Interface Card NIC
  - Part of P10 IDS/IPS system
  - We wrote our own firm and software for dynamic monitoring
- Front-end FPGA
  - Full line rate tap and repeater
- Tapped traffic
  - 128-bit wide words

System Overview



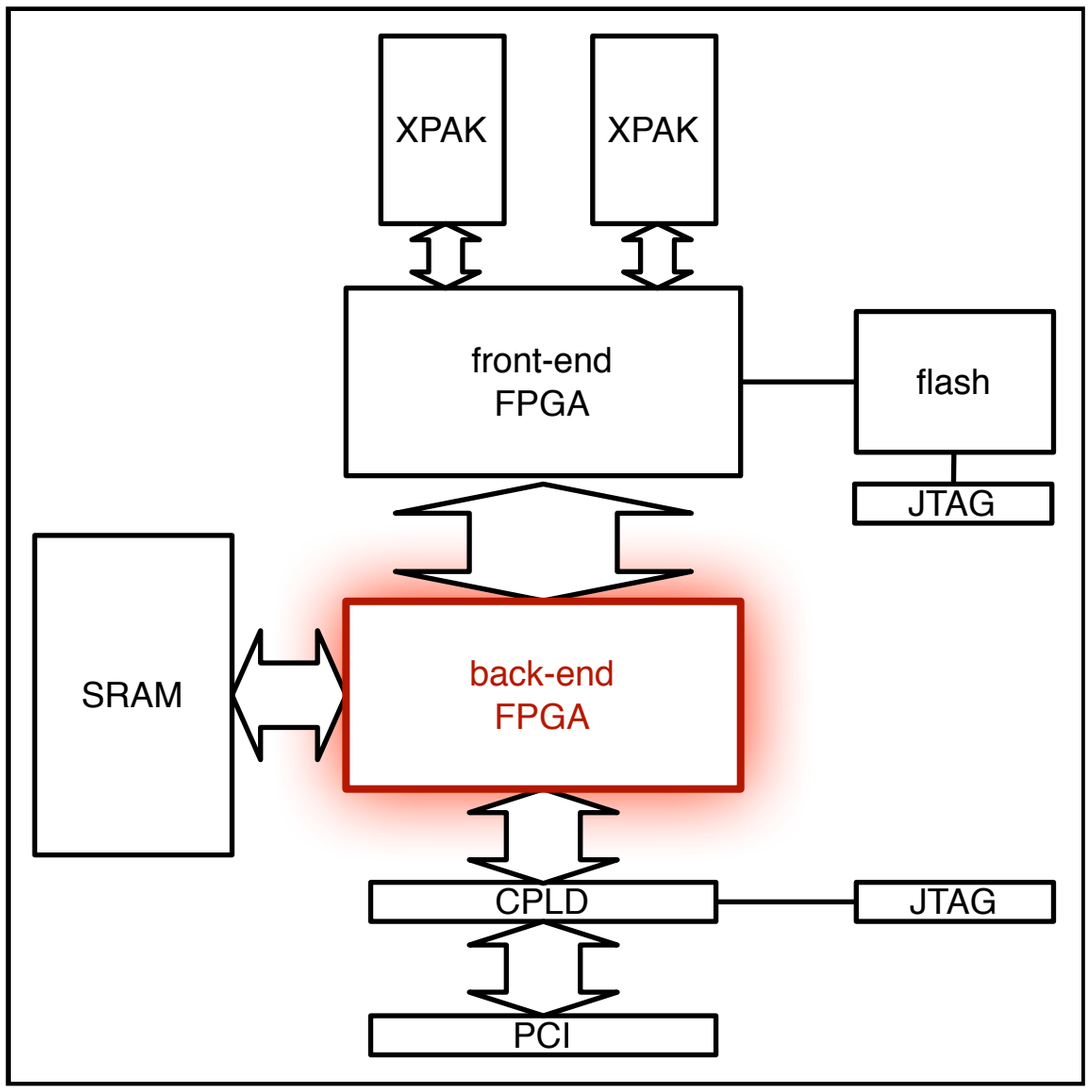
- Force10 MTP-10G
  - Network Interface Card NIC
  - Part of P10 IDS/IPS system
  - We wrote our own firm and software for dynamic monitoring
- Front-end FPGA
  - Full line rate tap and repeater
- Tapped traffic
  - 128-bit wide words

System Overview



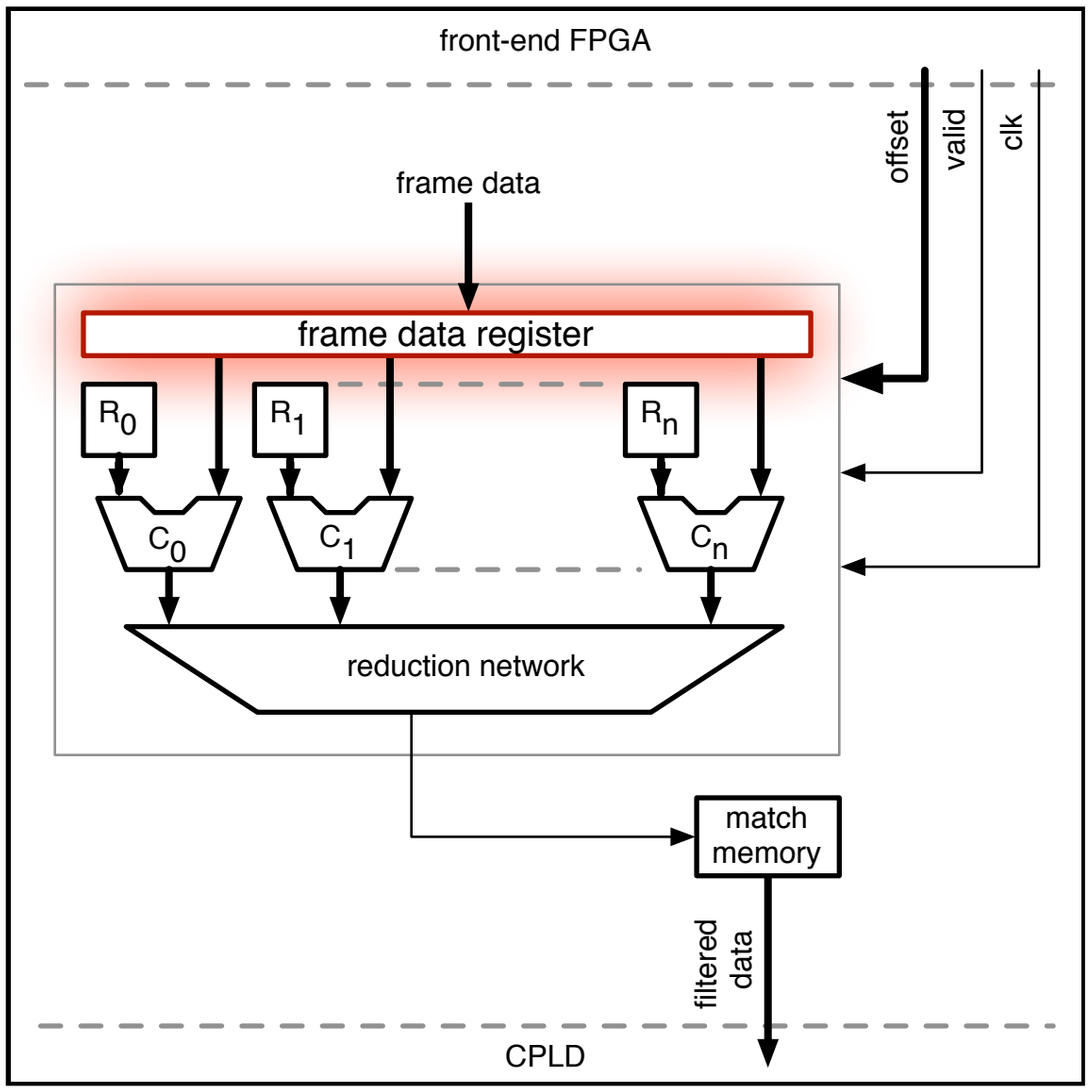
- Force10 MTP-10G
  - Network Interface Card NIC
  - Part of P10 IDS/IPS system
  - We wrote our own firm and software for dynamic monitoring
- Front-end FPGA
  - Full line rate tap and repeater
- Tapped traffic
  - 128-bit wide words

System Overview



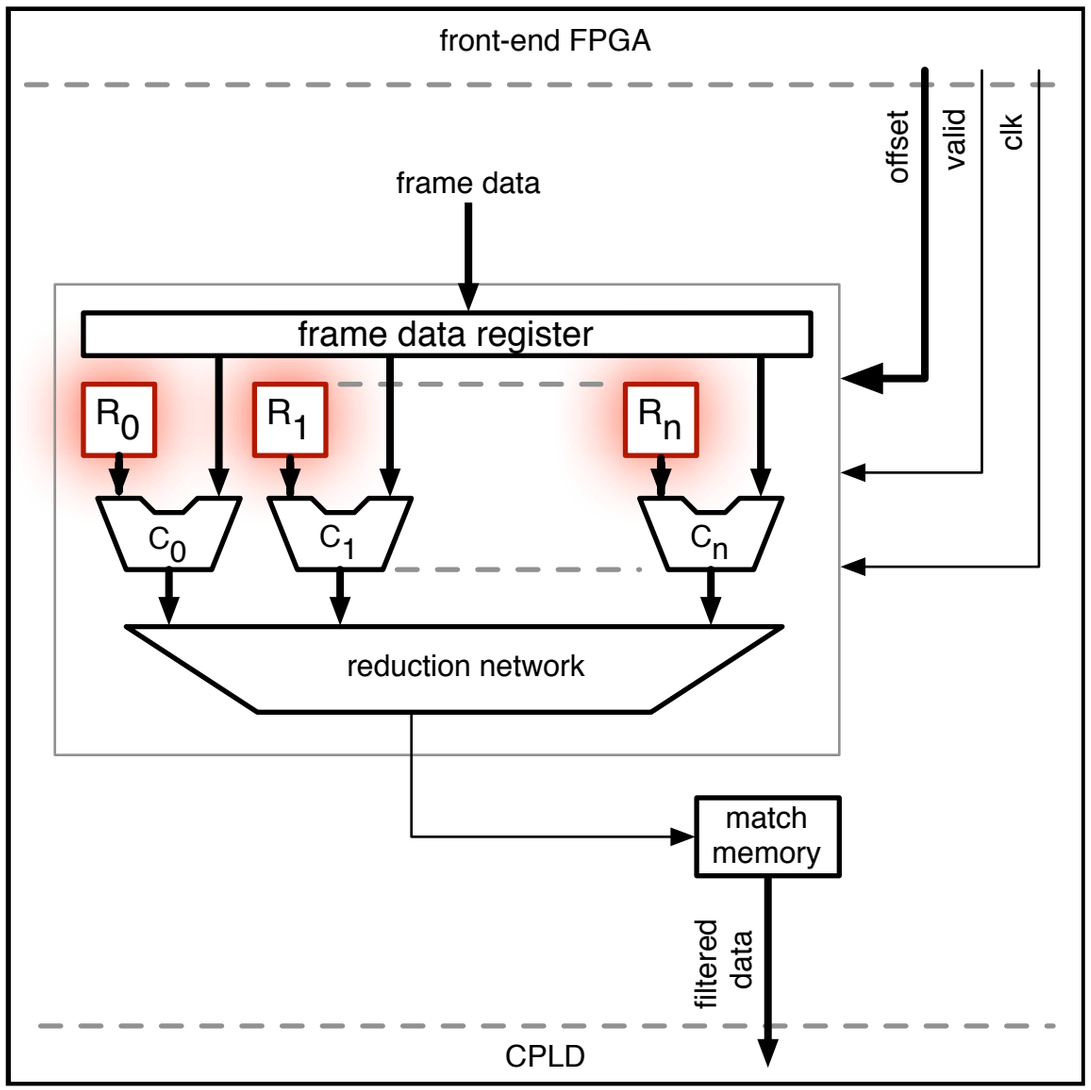
- Front-end FPGA
  - Full line rate tap and repeater
- Tapped traffic
  - 128-bit wide words
- Back-end FPGA
  - The heart of the system
  - MISD machine
  - Parallel processing on the 128-bit words.

System Overview



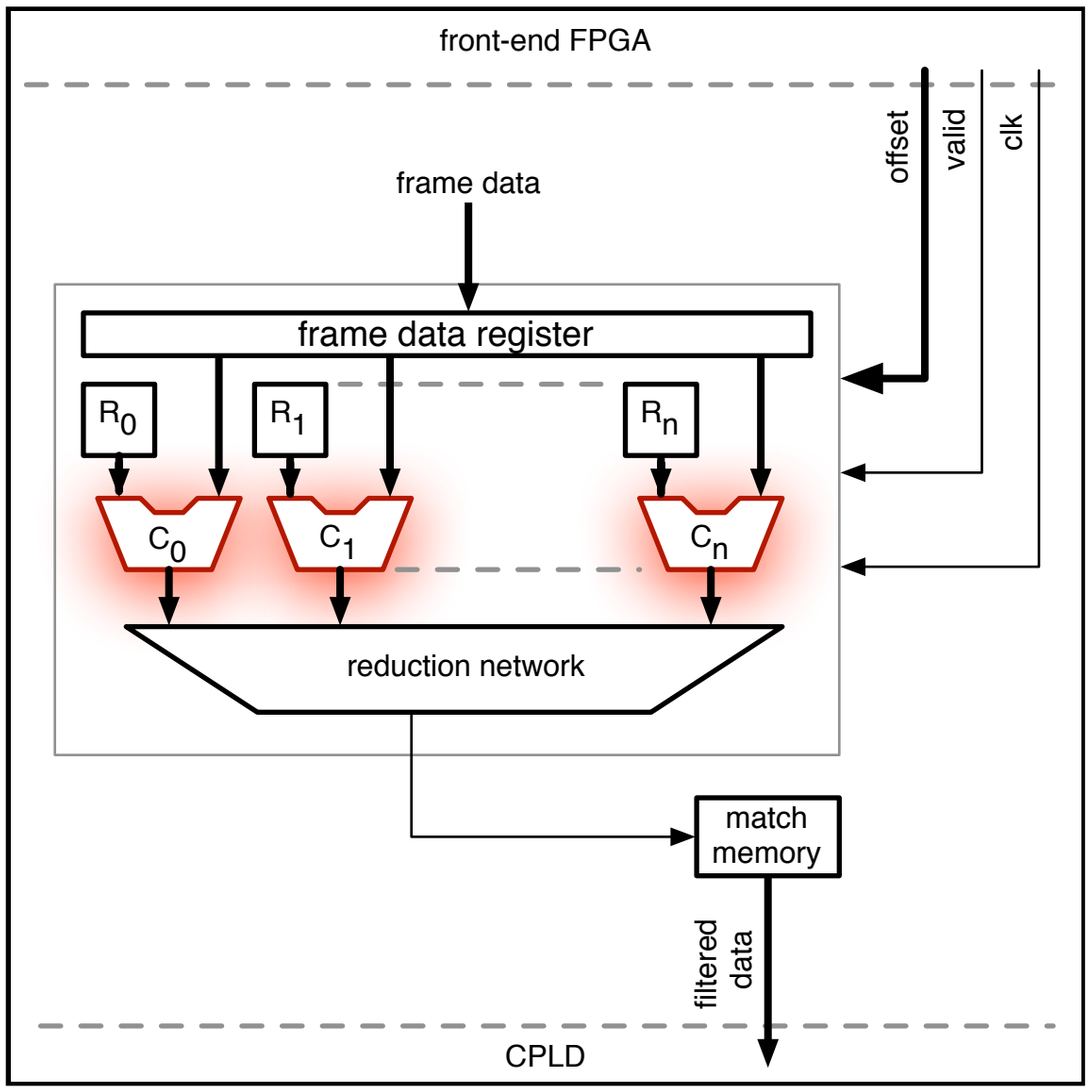
- Front-end FPGA
  - Full line rate tap and repeater
- Tapped traffic
  - 128-bit wide words
- Back-end FPGA
  - The heart of the system
  - MISD machine
  - Parallel processing on the 128-bit words.
- 128-bit word register (Single Data).

System Overview



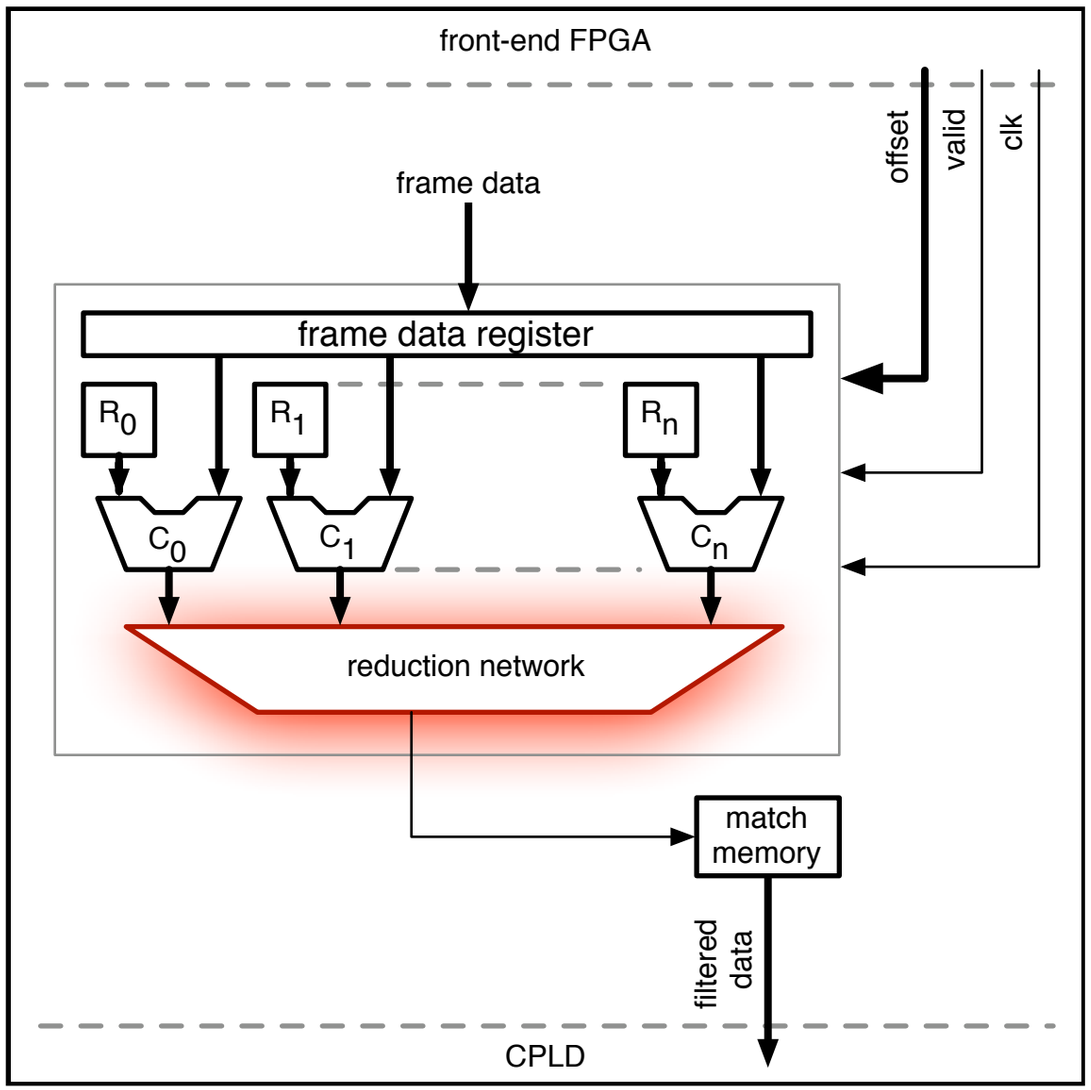
- Front-end FPGA
  - Full line rate tap and repeater
- Tapped traffic
  - 128-bit wide words
- Back-end FPGA
  - The heart of the system
  - MISD machine
  - Parallel processing on the 128-bit words.
- 128-bit word register (Single Data).
- Multiple instruction registers.

System Overview



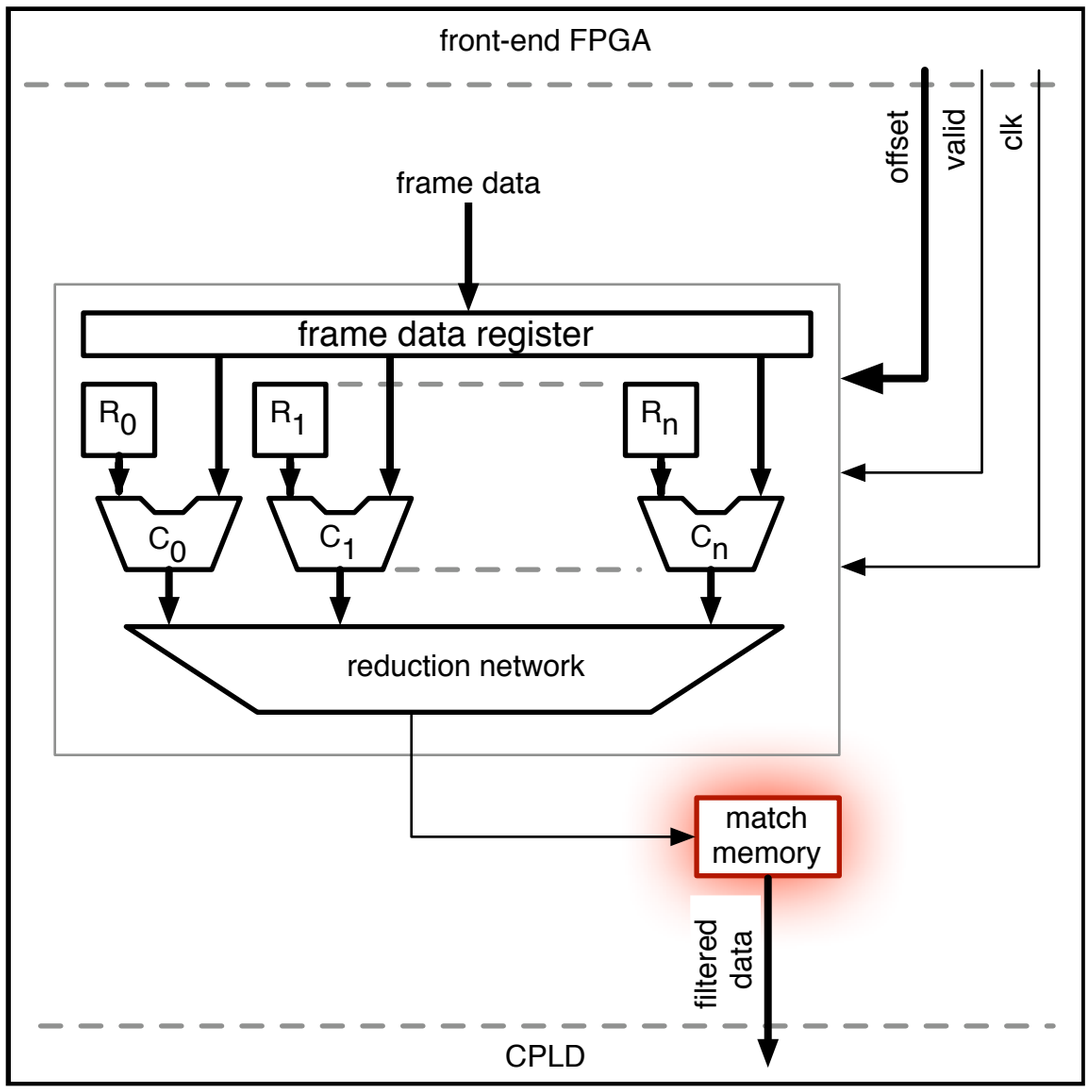
- Back-end FPGA
  - The heart of the system
  - MISD machine
  - Parallel processing on the 128-bit words.
- 128-bit word register (Single Data).
- Multiple instruction registers.
- Multiple processors.

System Overview



- Back-end FPGA
  - The heart of the system
  - MISD machine
  - Parallel processing on the 128-bit words.
- 128-bit word register (Single Data).
- Multiple instruction registers.
- Multiple processors.
  - Result is reduced to a match or no-match signal.

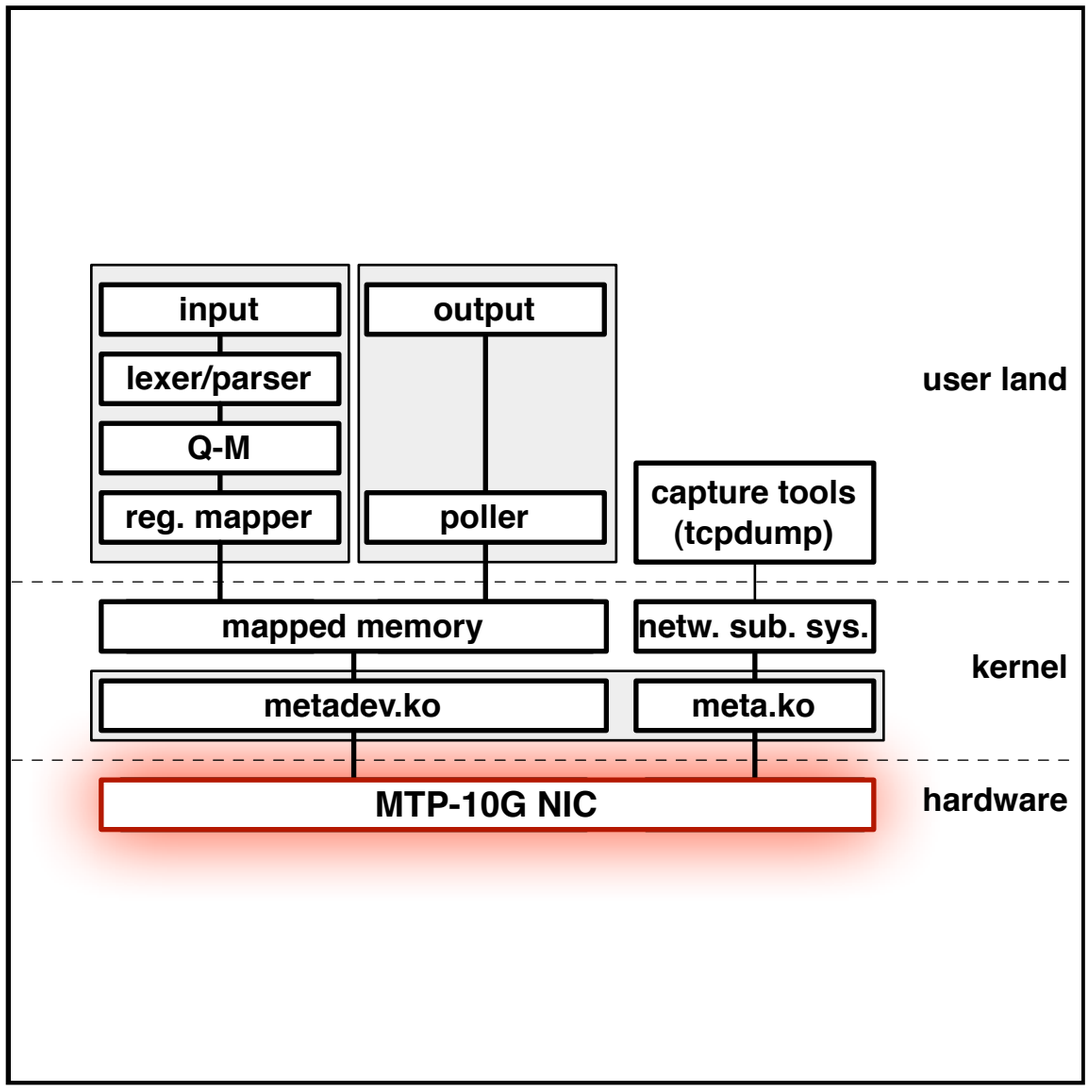
System Overview



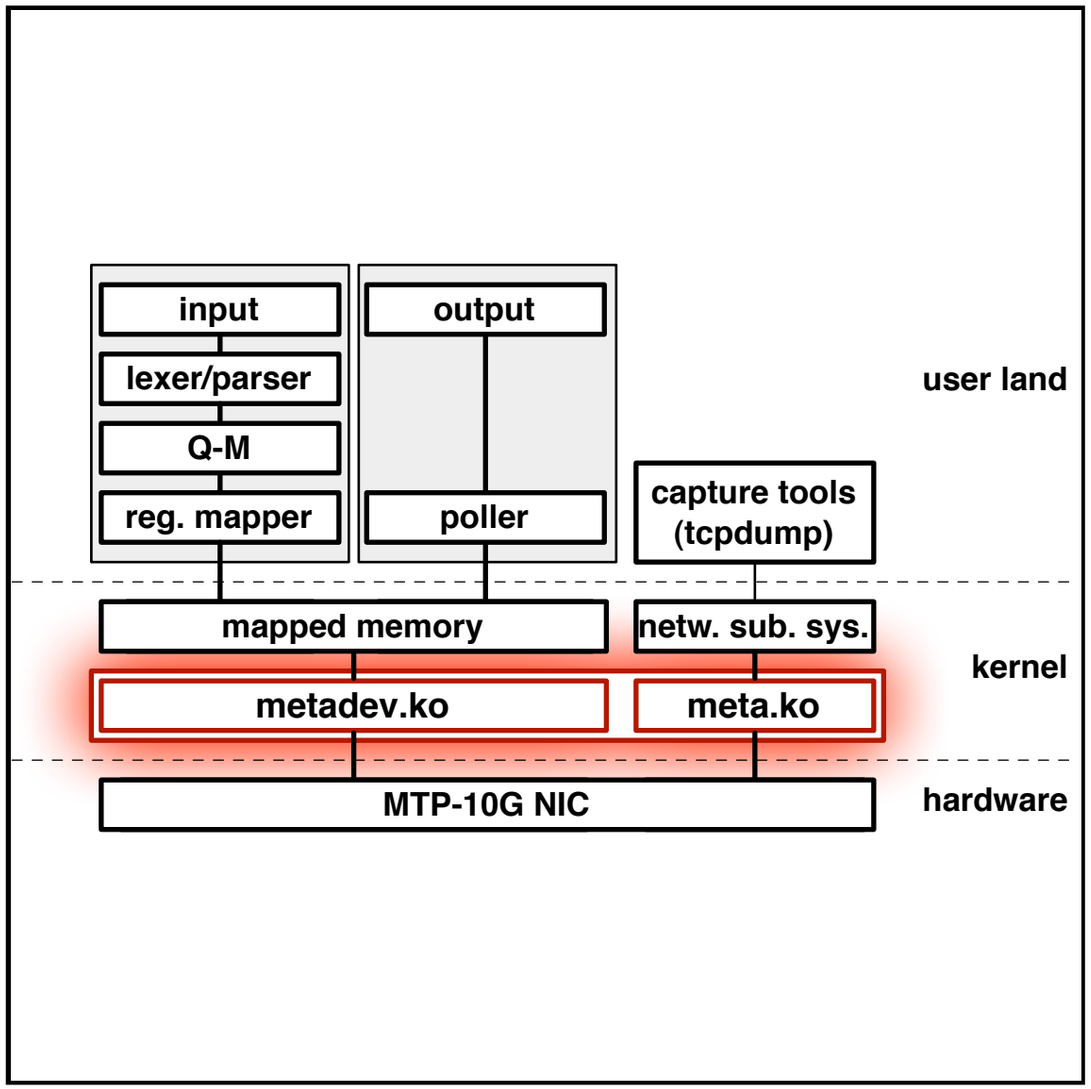
- Back-end FPGA
  - The heart of the system
  - MISD machine
  - Parallel processing on the 128-bit words.
- 128-bit word register (Single Data).
- Multiple instruction registers.
- Multiple processors.
  - Result is reduced to a match or no-match signal.
  - Frame is copied down to the PCI-X interface upon a match.

System Overview

- Hardware

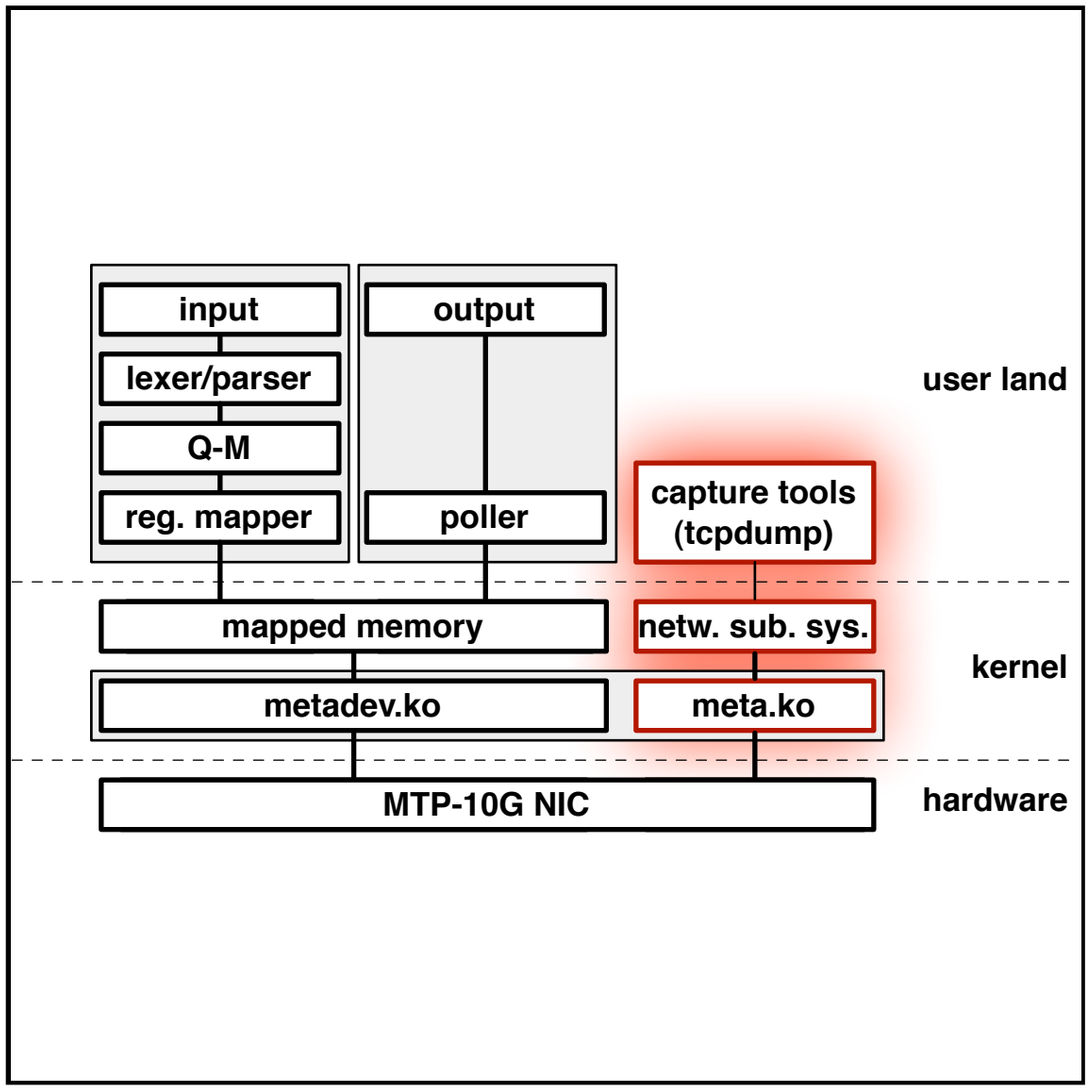


System Overview



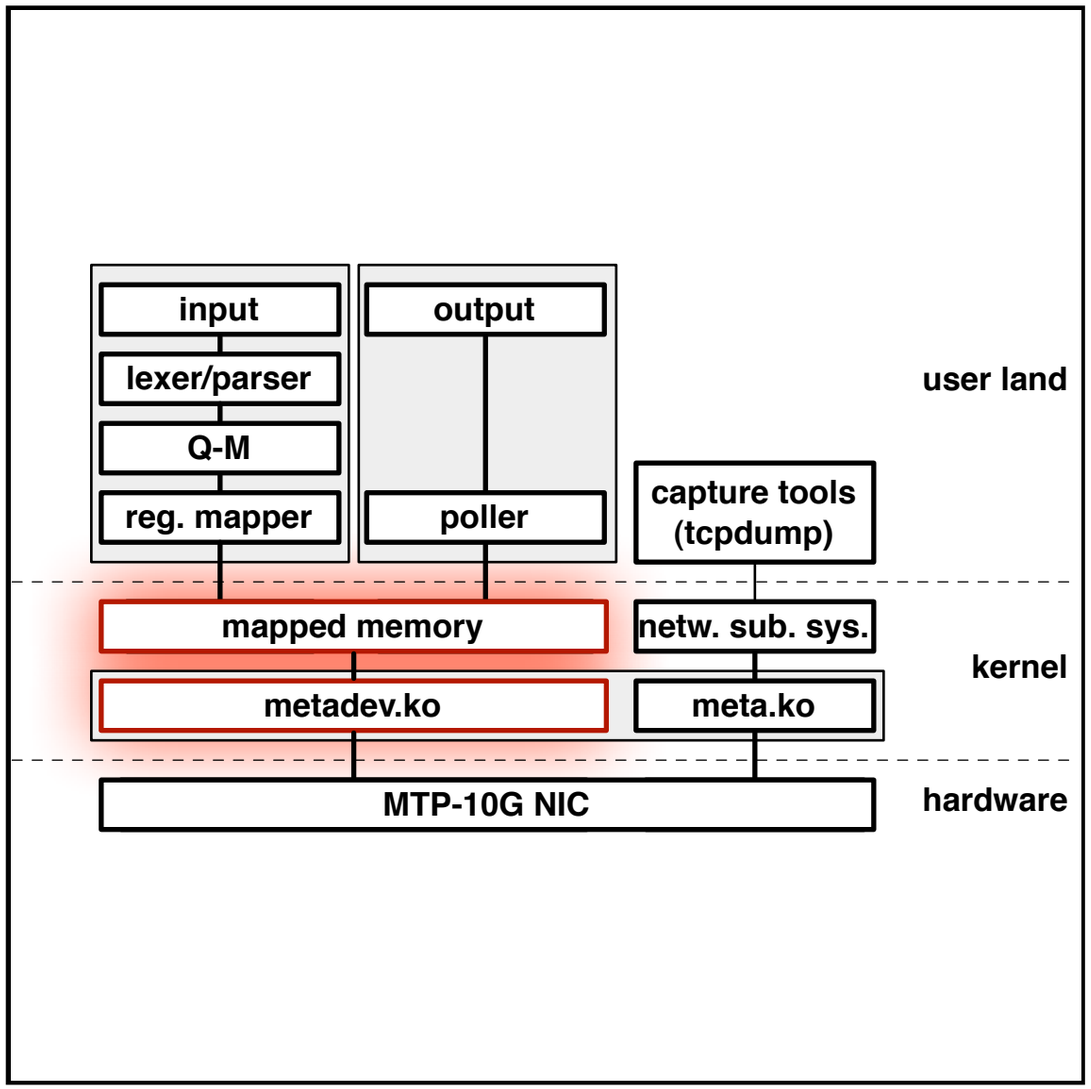
- Hardware
- Software
  - Two kernel modules.

System Overview



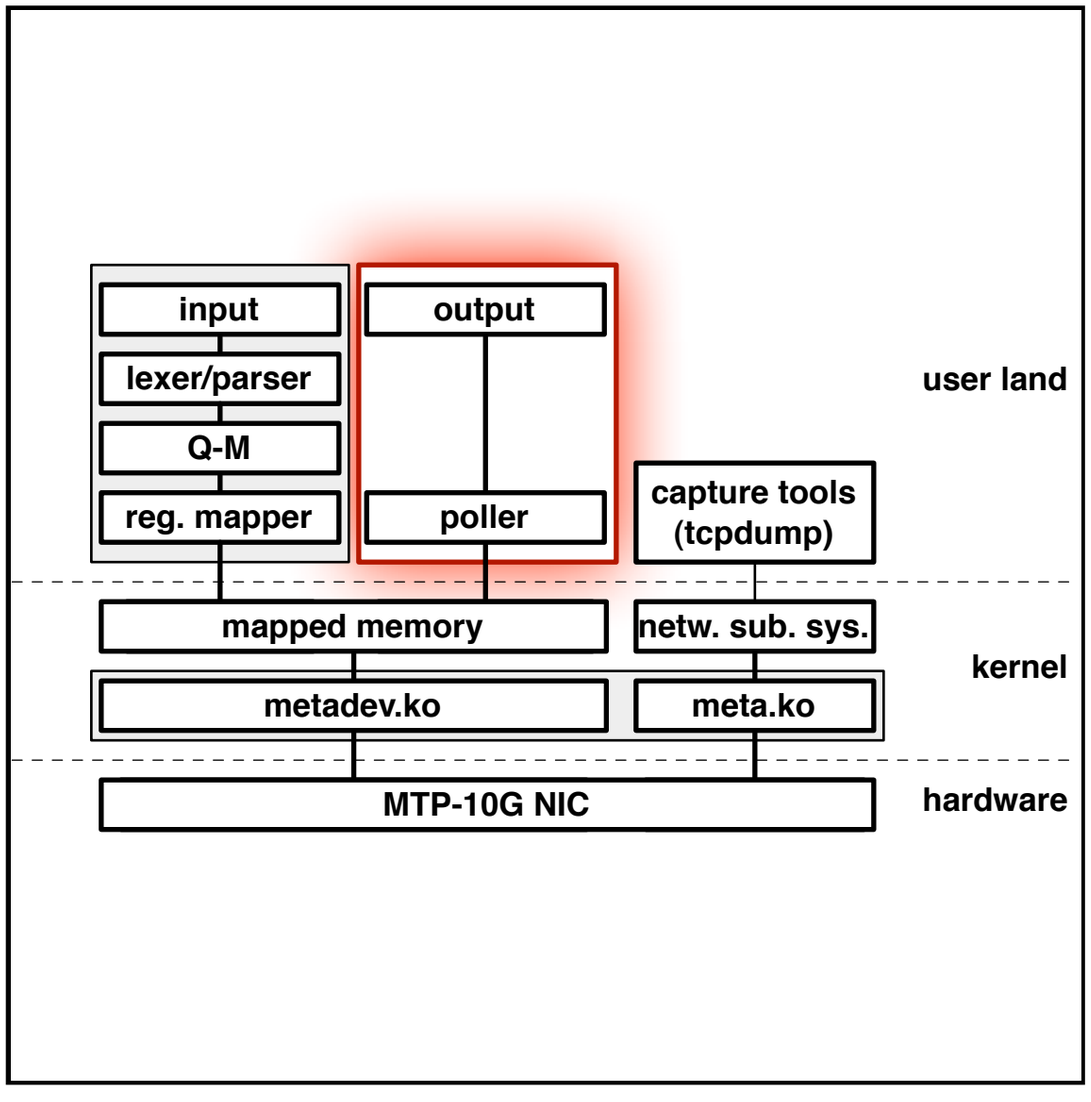
- Hardware
- Software
  - Two kernel modules.
    - Network subsystem
    - libpcap

System Overview



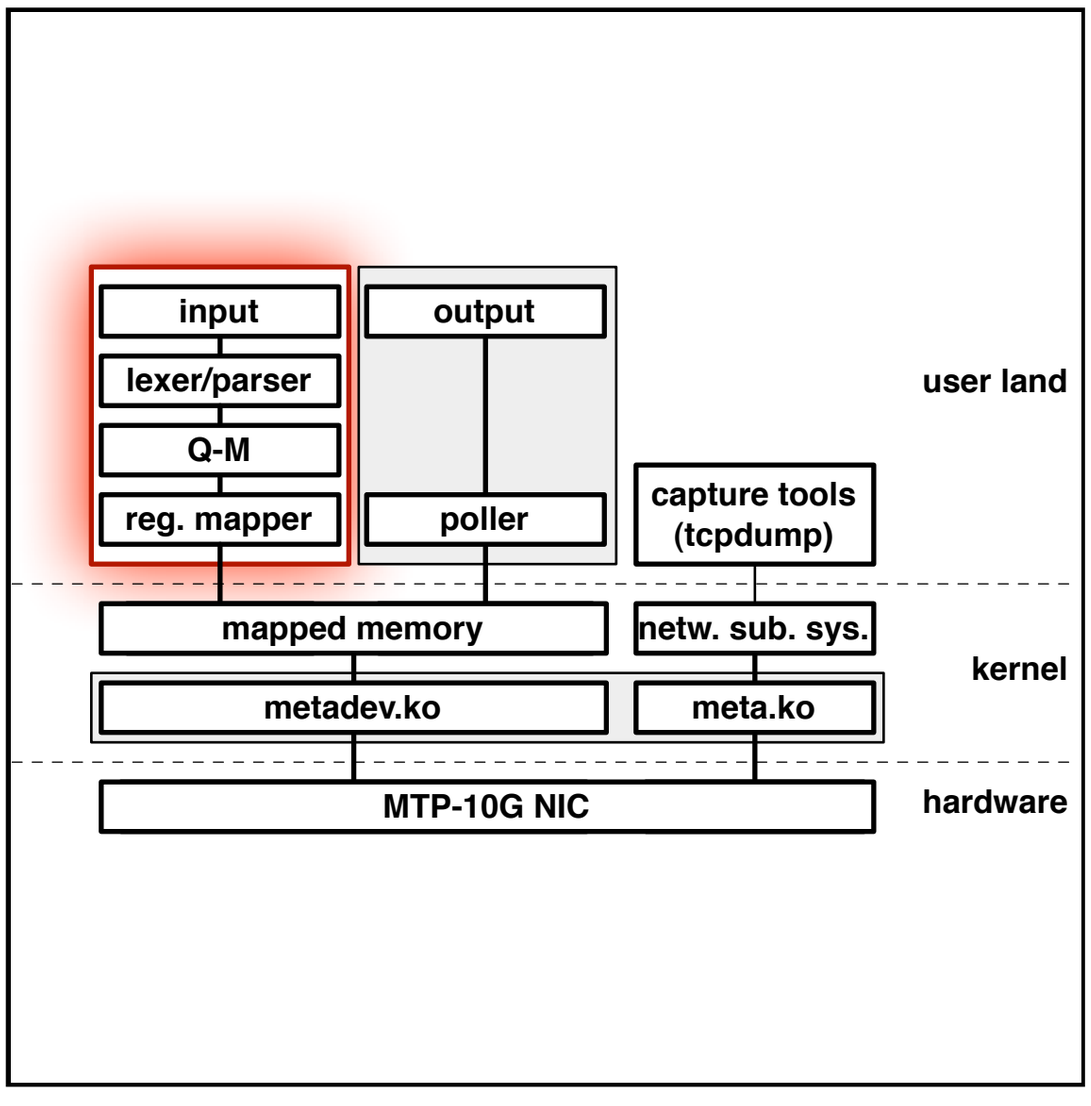
- Hardware
- Software
  - Two kernel modules.
    - Network subsystem
      - libpcap
    - Mapped memory
      - I/O

System Overview



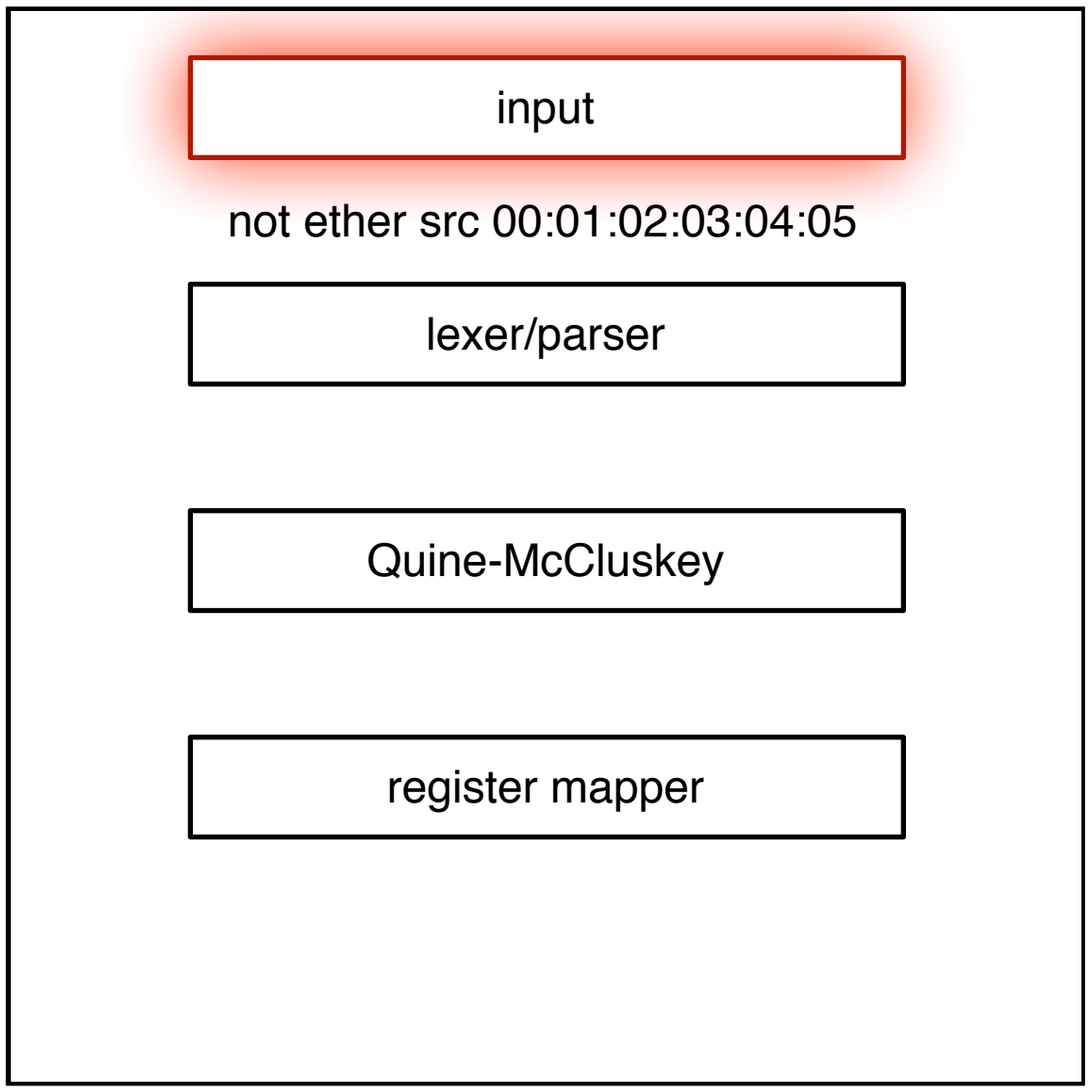
- Hardware
- Software
  - Two kernel modules.
    - Network subsystem
      - libpcap
    - Mapped memory
      - I/O
      - Counter readings

System Overview



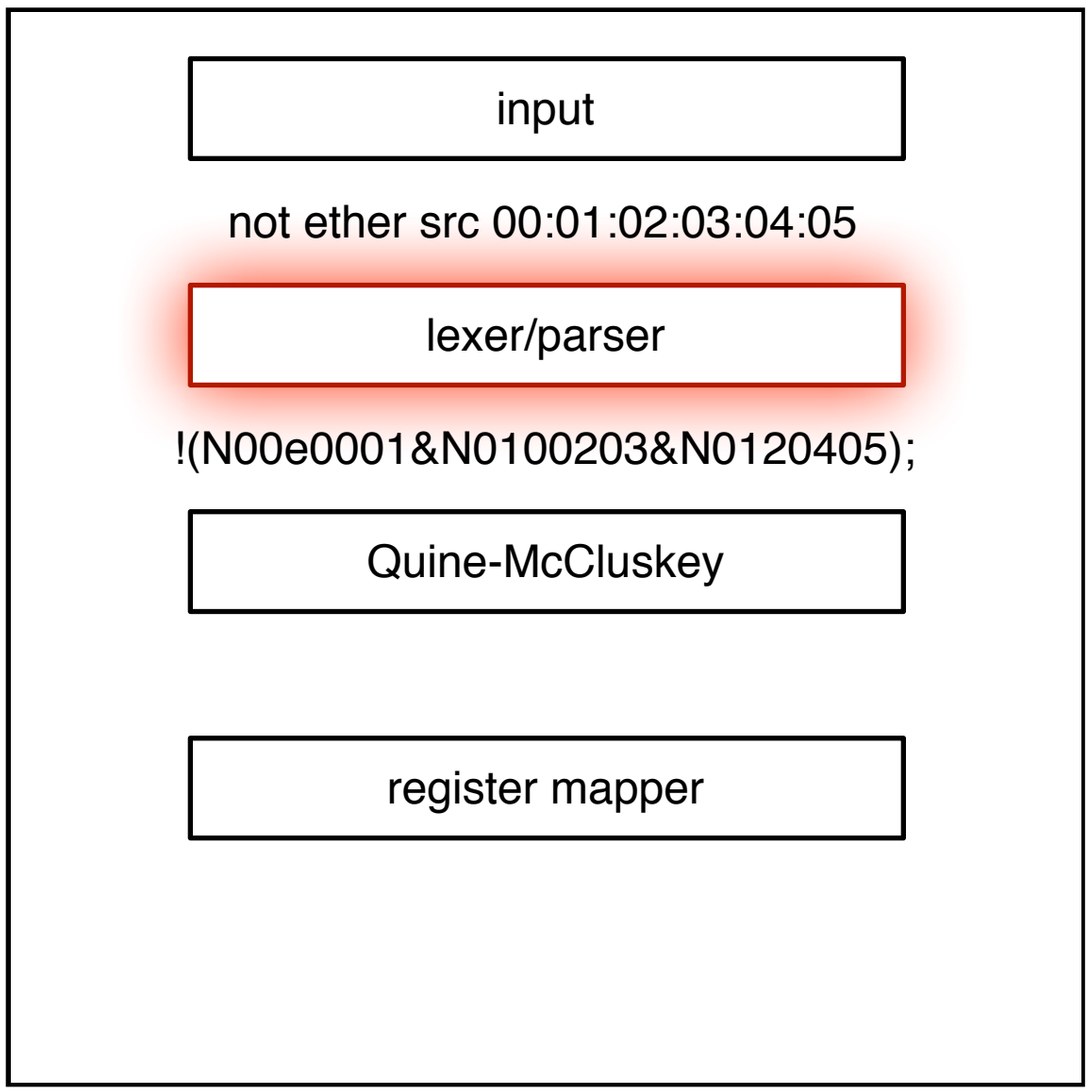
- Hardware
- Software
  - Two kernel modules.
    - Network subsystem
      - libpcap
    - Mapped memory
      - I/O
      - Counter readings
      - Dynamic filter settings

System Overview



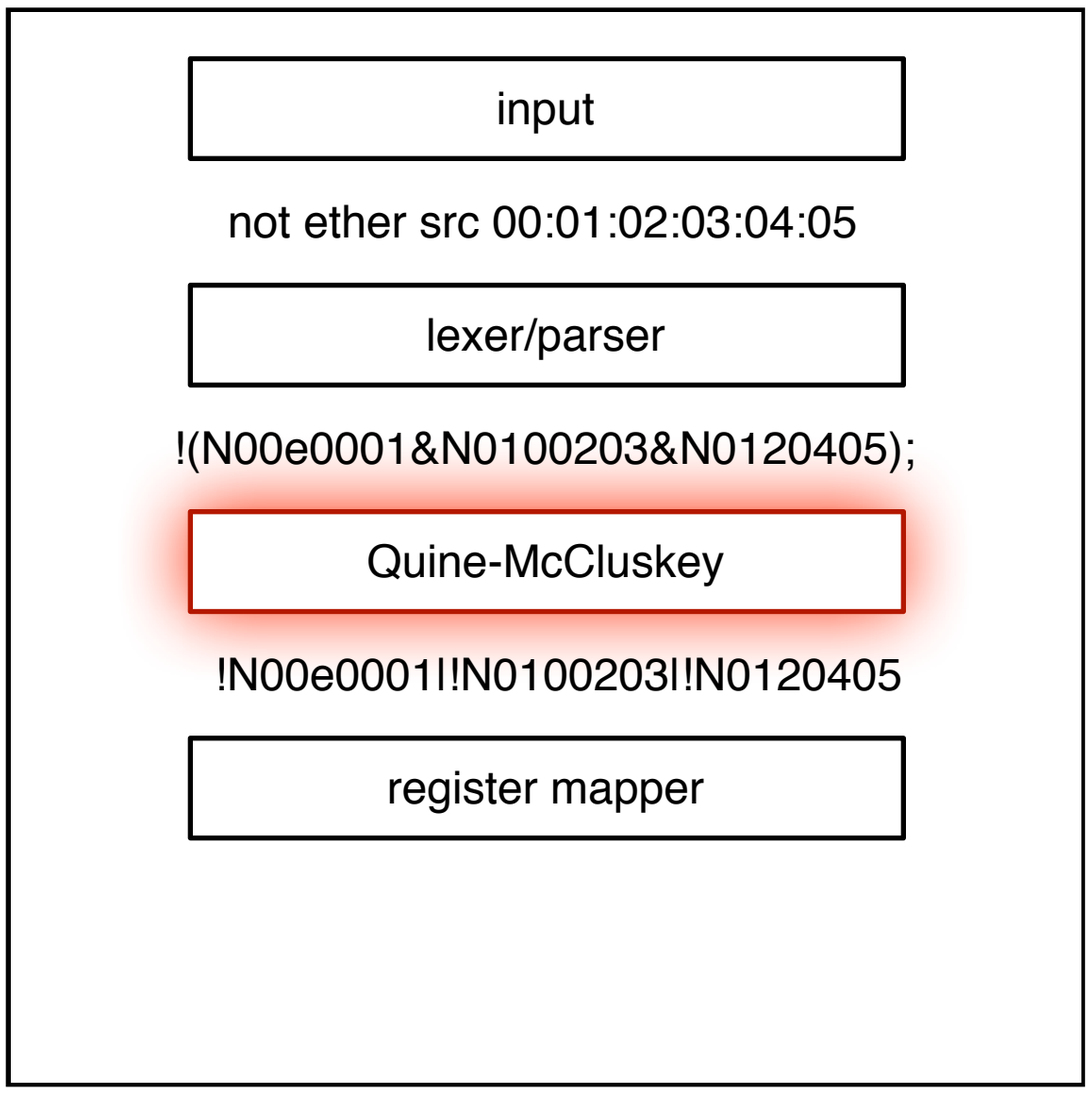
- Hardware
- Software part
  - Two kernel modules.
    - Network subsystem
      - libpcap
    - Mapped memory
      - I/O
      - Counter readings
      - Dynamic filter settings.
- Takes input from stdin.

System Overview



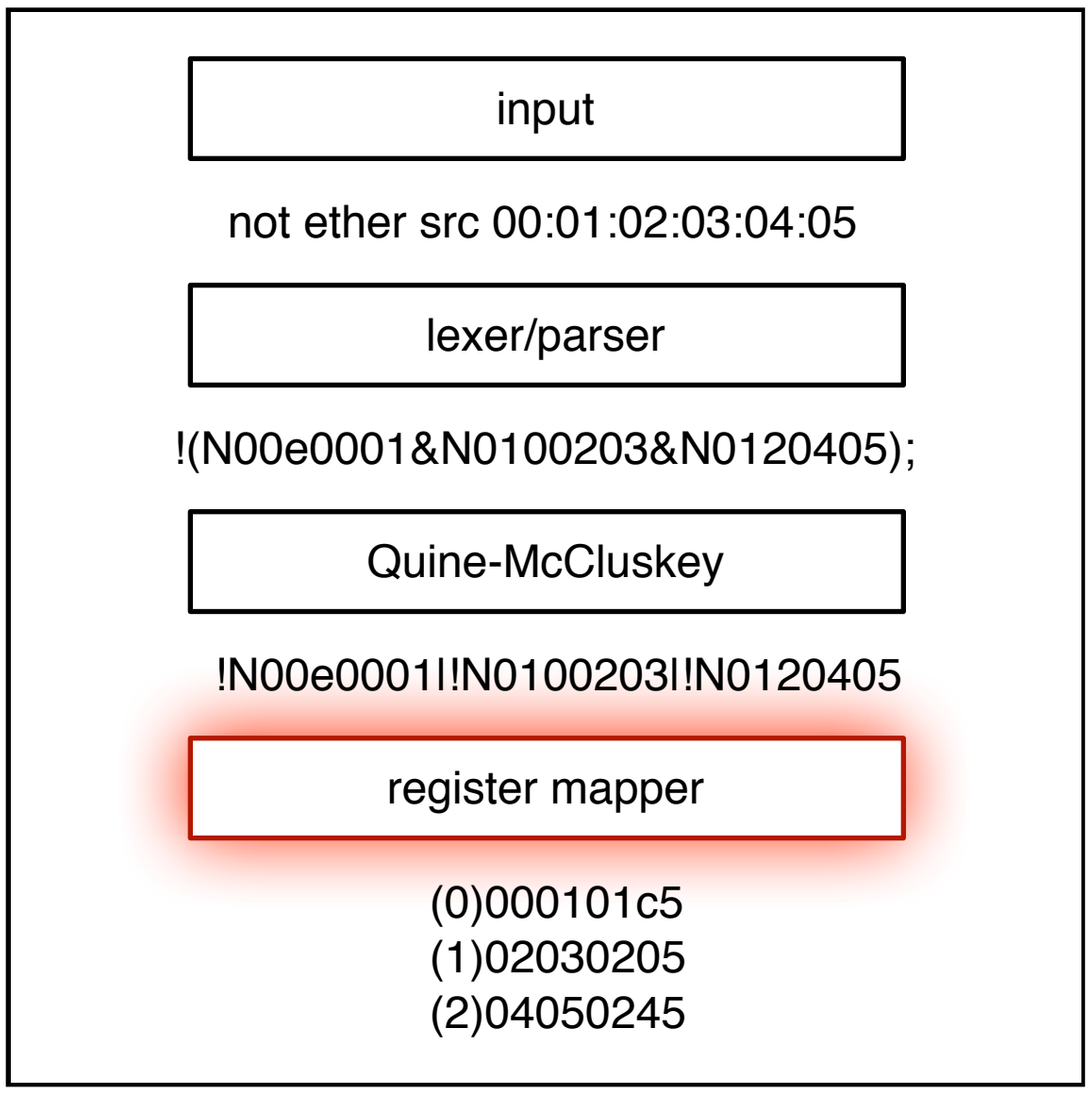
- Takes input from stdin.
- Parses the input.
  - Chops addresses in patterns and offsets.
  - Results in an internal expression representation.

System Overview



- Takes input from stdin.
- Parses the input.
  - Chops addresses in patterns and offsets.
  - Results in an internal expression representation.
- Quine-McCluskey
  - Transforms the expression into its disjunctive normal form.

System Overview



- Takes input from stdin.
- Parses the input.
  - Chops addresses in patterns and offsets.
  - Results in an internal expression representation.
- Quine-McCluskey
  - Transforms the expression into its disjunctive normal form.
- Maps each term into a instruction register in the back-end FPGA.

Agenda

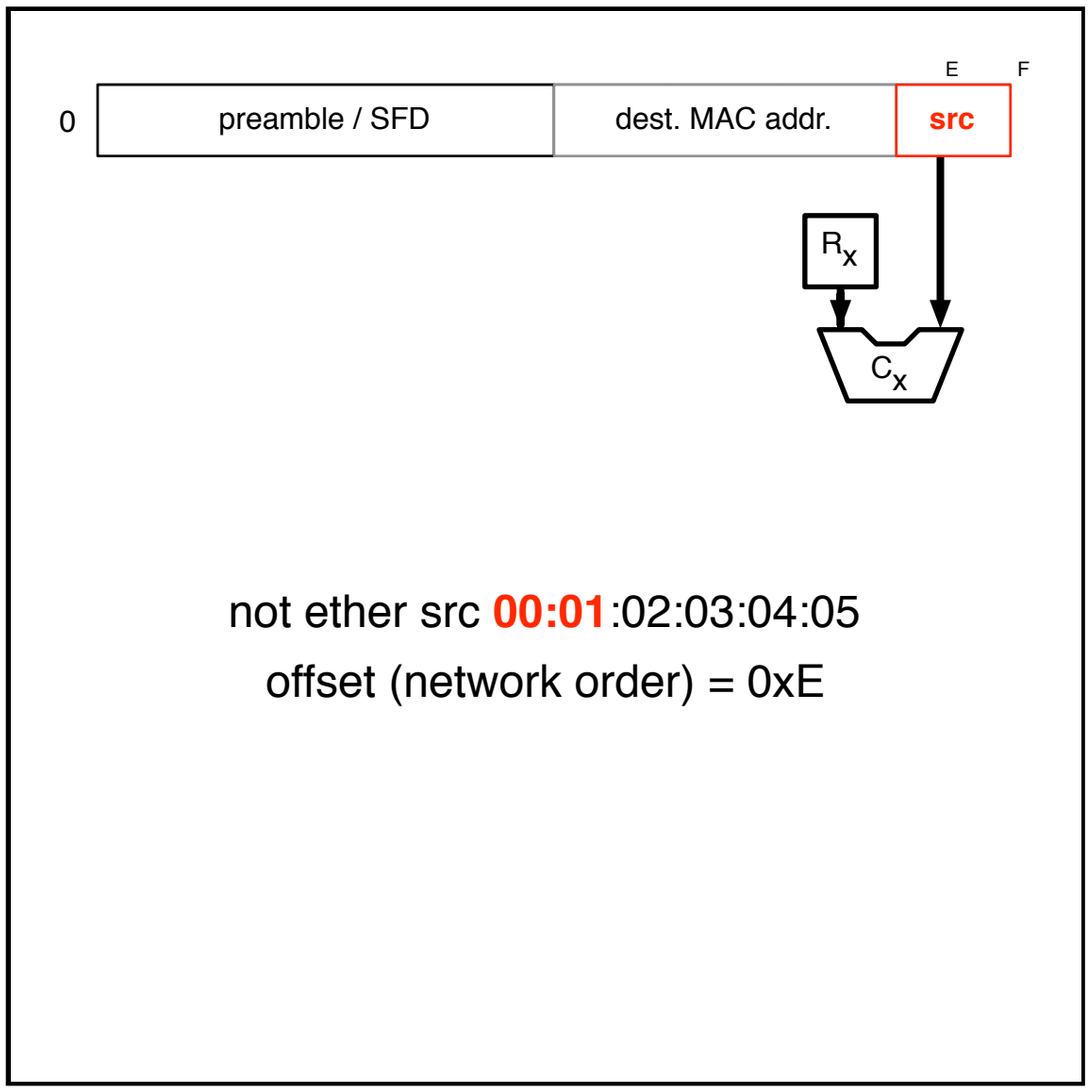
- An example.
- The problems with passive 10GbE monitoring.
  - Performance of generic computer hardware.
  - Issues with the mirror port feature.
- Overview of our solution.
  - Hardware architecture.
  - Homegrown firmware and software.
- A bit more detail.
- More to demo.

Details

- Let's see how this filter expression is handled in detail.

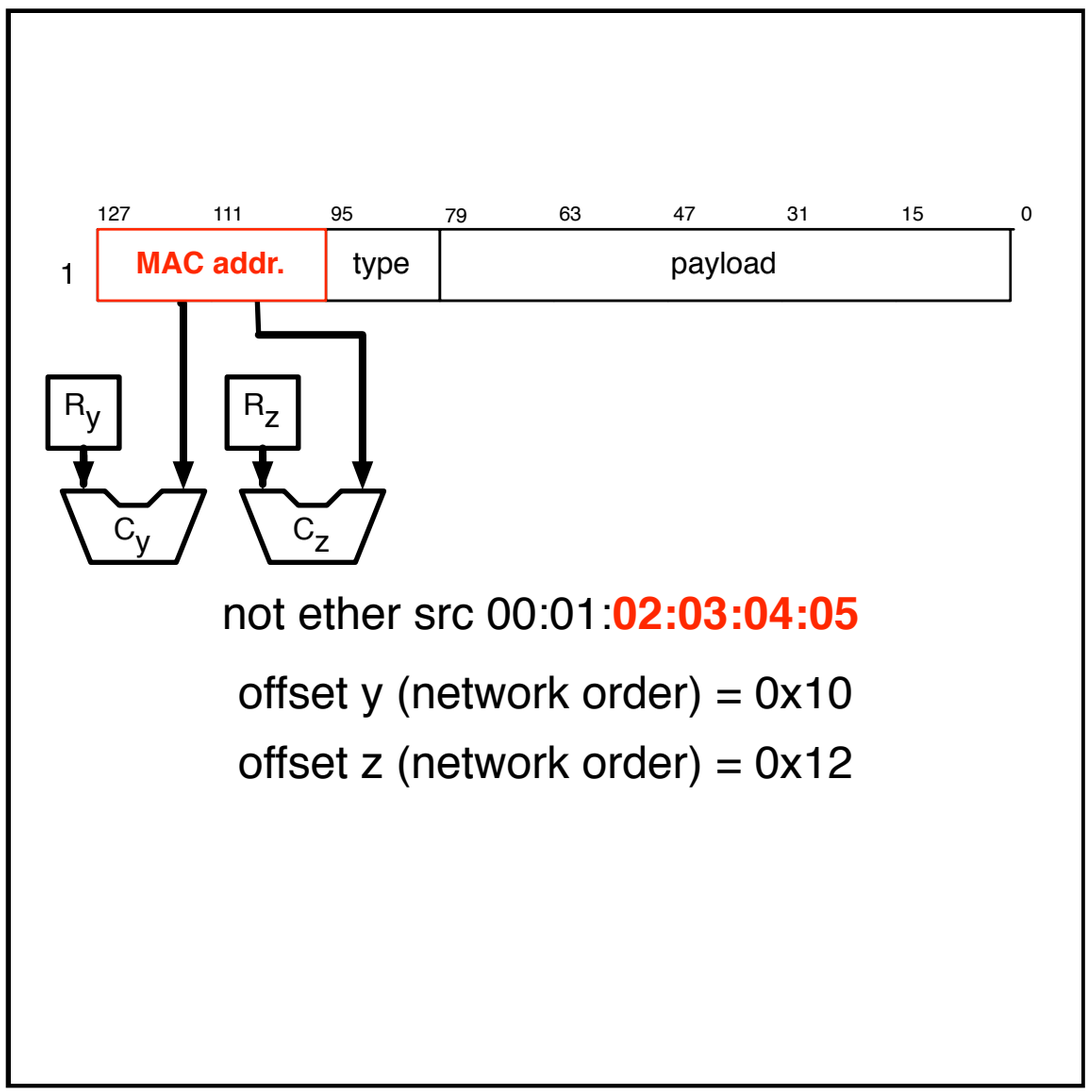
not ether src 00:01:02:03:04:05

Details



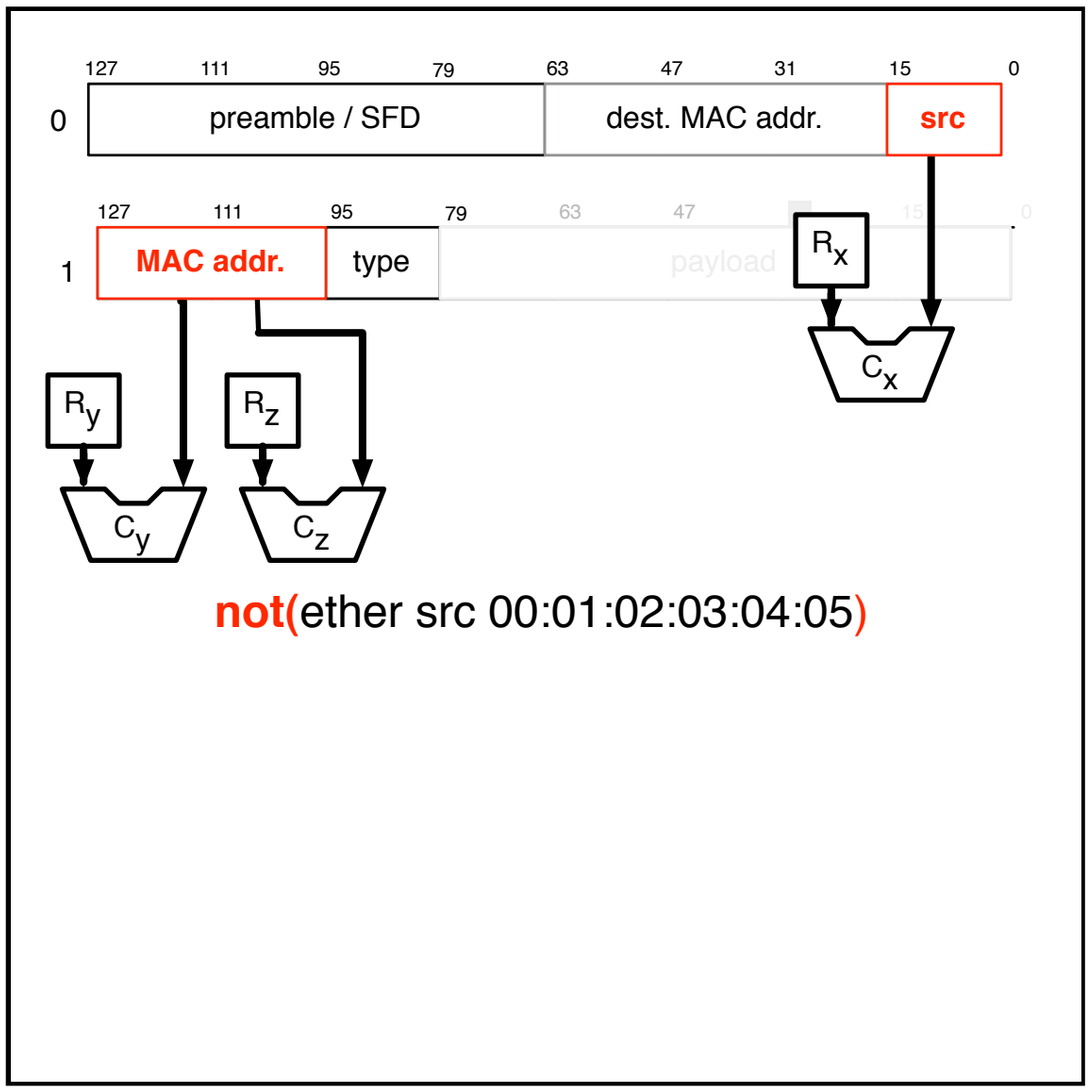
- Let's see how this filter expression is handled in detail.
- The first 128bit word contains the first two bytes of the ethernet source address.

Details



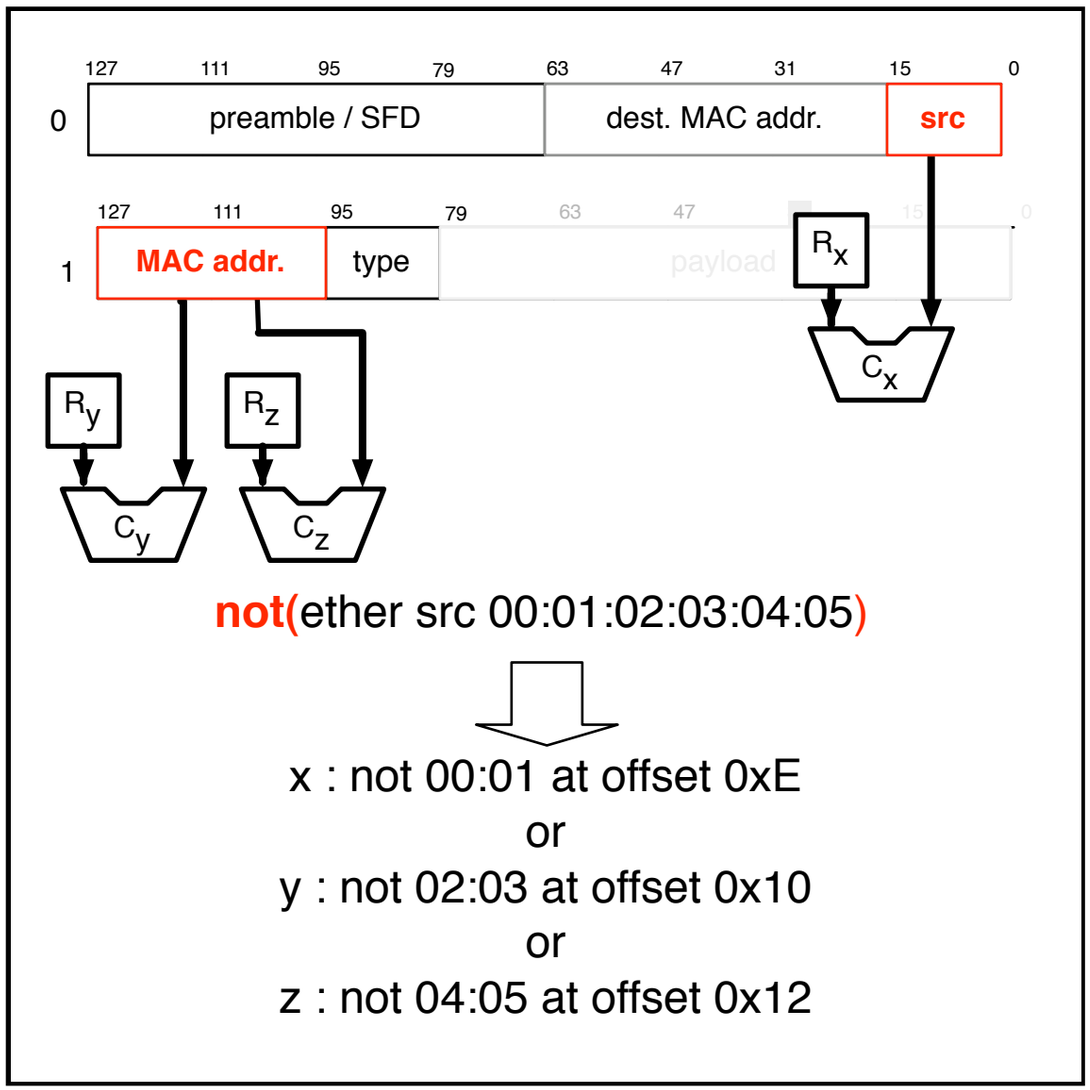
- Let's see how this filter expression is handled in detail.
- The first 128bit word contains the first two bytes of the ethernet source address.
- The second 128bit word contains the remaining address.
  - One "processor" can process two bytes.
  - Hence we need two here.

Details



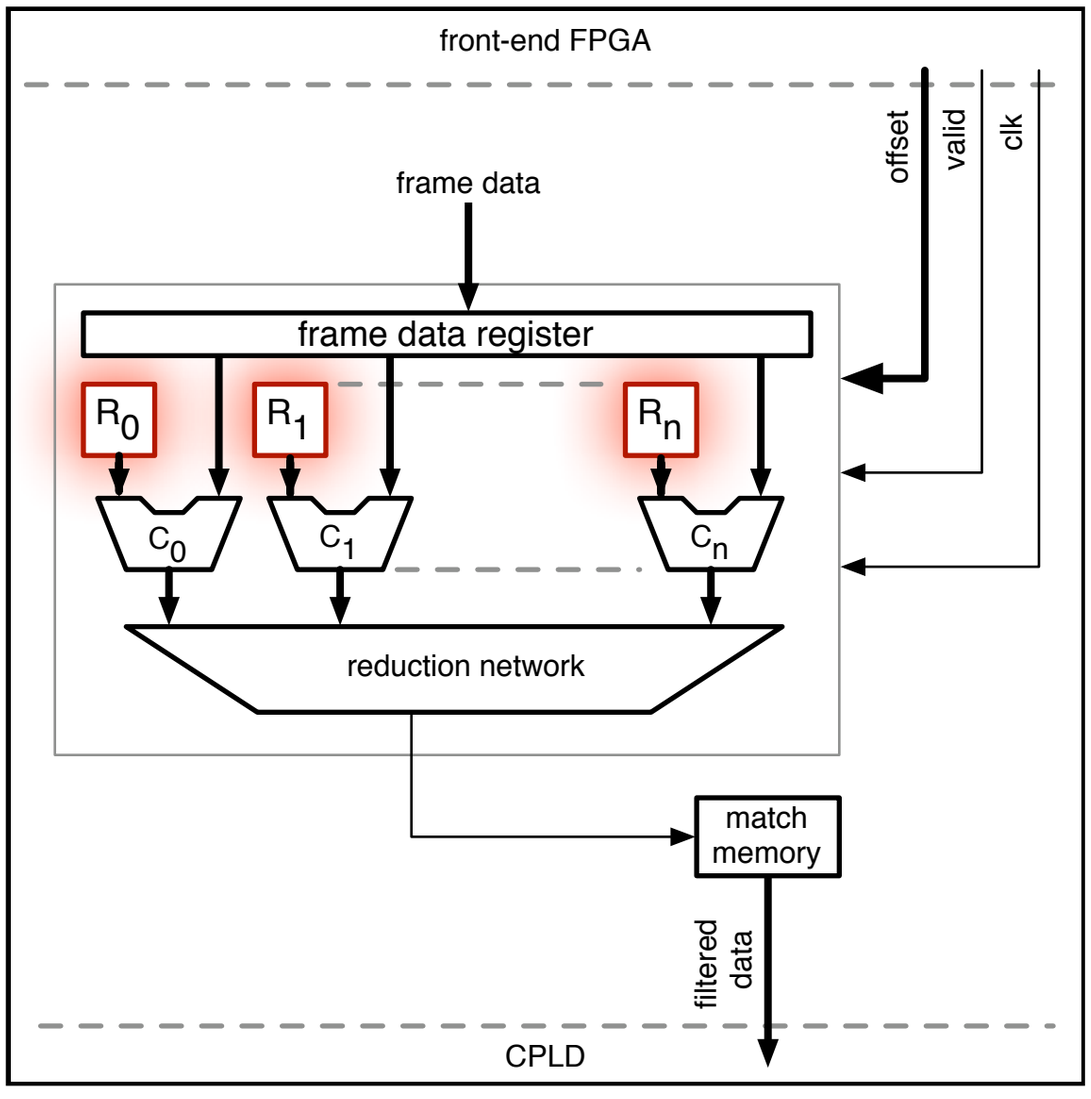
- So far so good, but what about this not operator?
  - Programmable network behind processors?
    - Limited in both speed and scalability.

Details



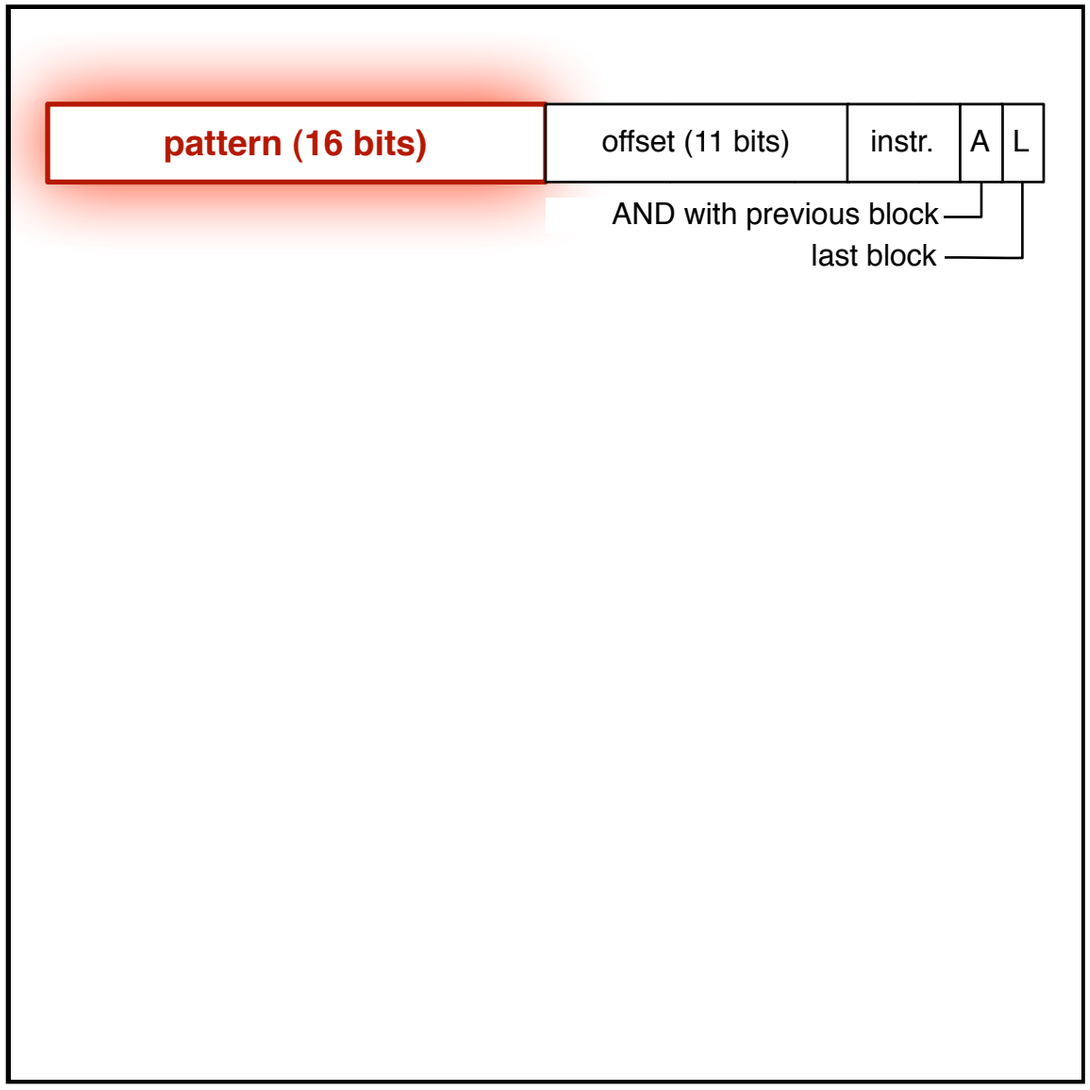
- So far so good, but what about this not operator?
  - Programmable network behind processors?
    - Limited in both speed and scalability.
- Do it is software by transforming the expression into sum of products
  - Disjunctive Normal Form.
  - Quine-McCluskey
    - Library

Details



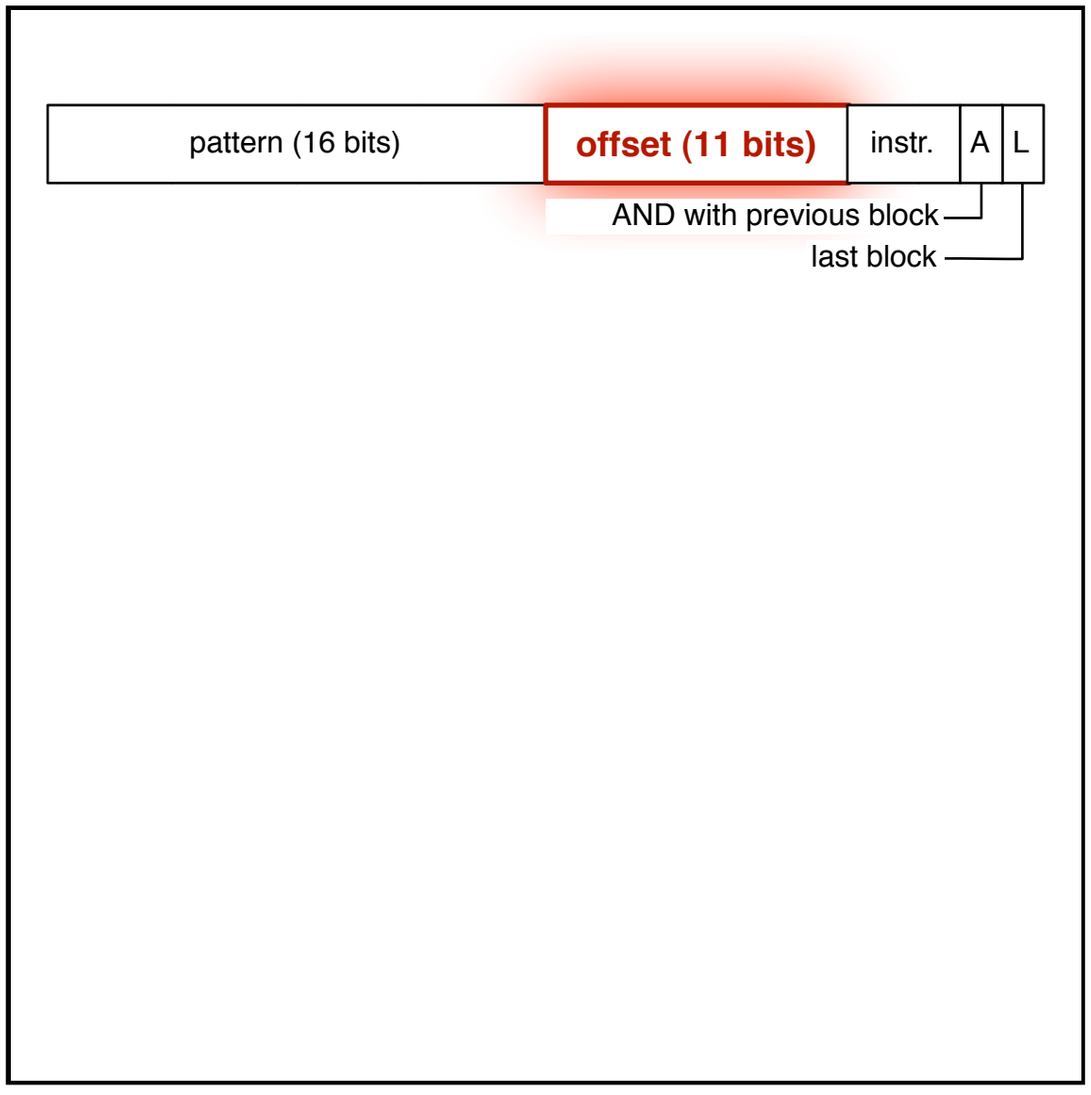
- So far so good, but what about this not operator?
  - Programmable network behind processors?
    - Limited in both speed and scalability.
  - Do it is software by transforming the expression into sum of products
    - Disjunctive Normal Form.
    - Quine-McCluskey
      - Library
- Registers

Details



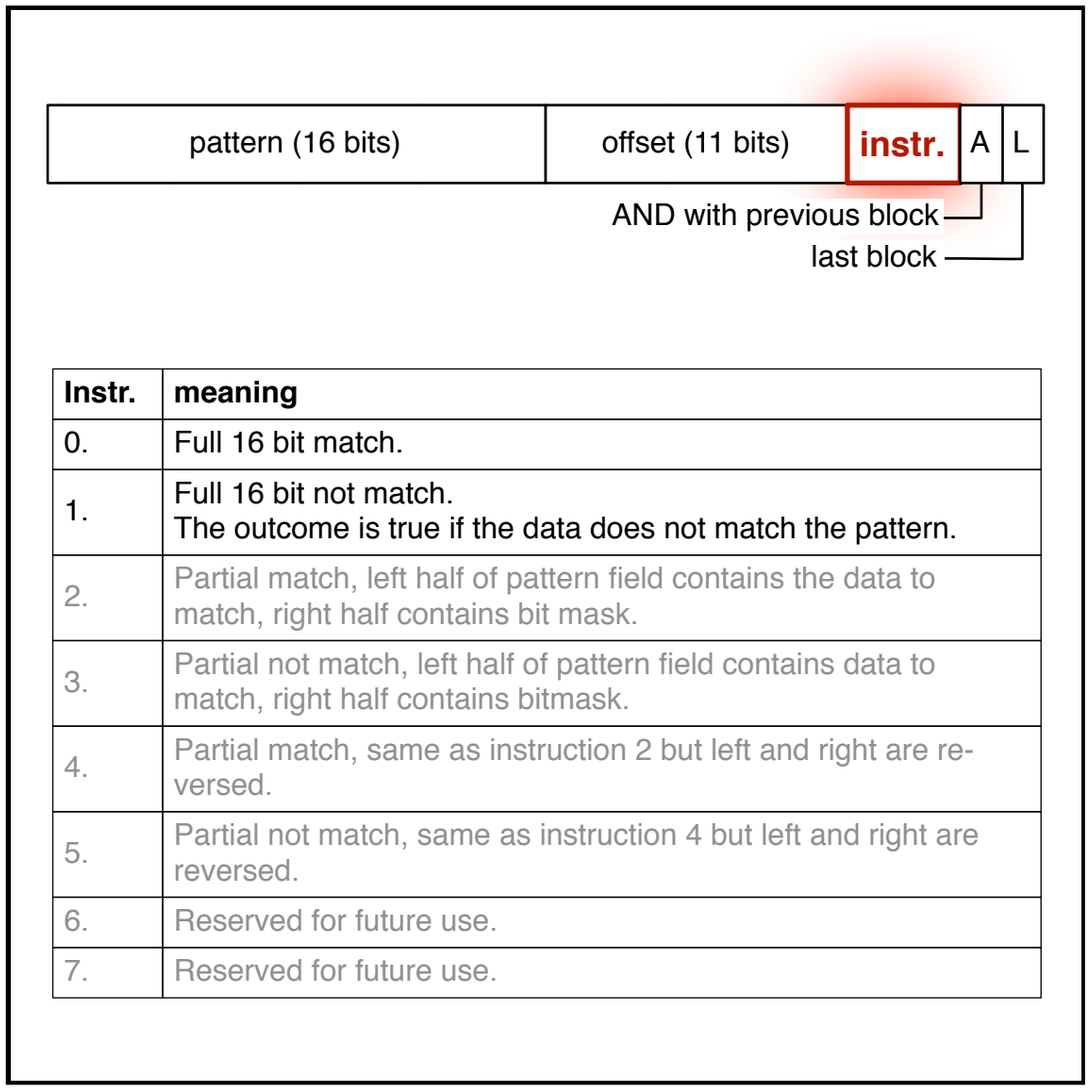
- Registers
  - Contain the pattern to match (16 bits)

Details



- Registers
  - Contain the pattern to match (16 bits)
  - The offset (11 bits)
    - 2048 bytes
    - 1522 byte frames

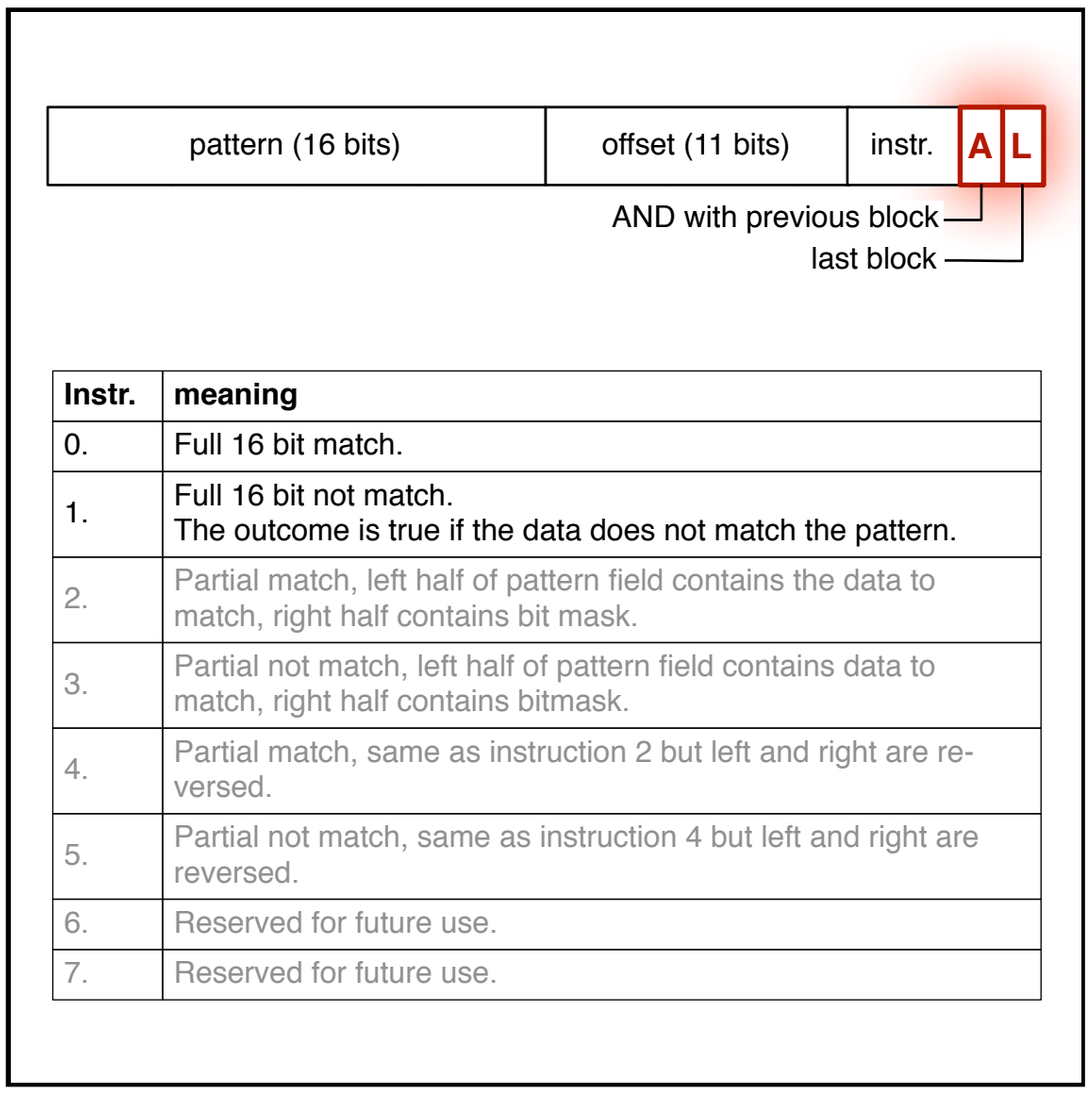
Details



Instr.	meaning
0.	Full 16 bit match.
1.	Full 16 bit not match. The outcome is true if the data does not match the pattern.
2.	Partial match, left half of pattern field contains the data to match, right half contains bit mask.
3.	Partial not match, left half of pattern field contains data to match, right half contains bitmask.
4.	Partial match, same as instruction 2 but left and right are re-versed.
5.	Partial not match, same as instruction 4 but left and right are reversed.
6.	Reserved for future use.
7.	Reserved for future use.

- Registers
  - Contain the pattern to match (16 bits)
  - The offset (11 bits)
    - 2048 bytes
    - 1522 byte frames
- Instruction (3 bits)

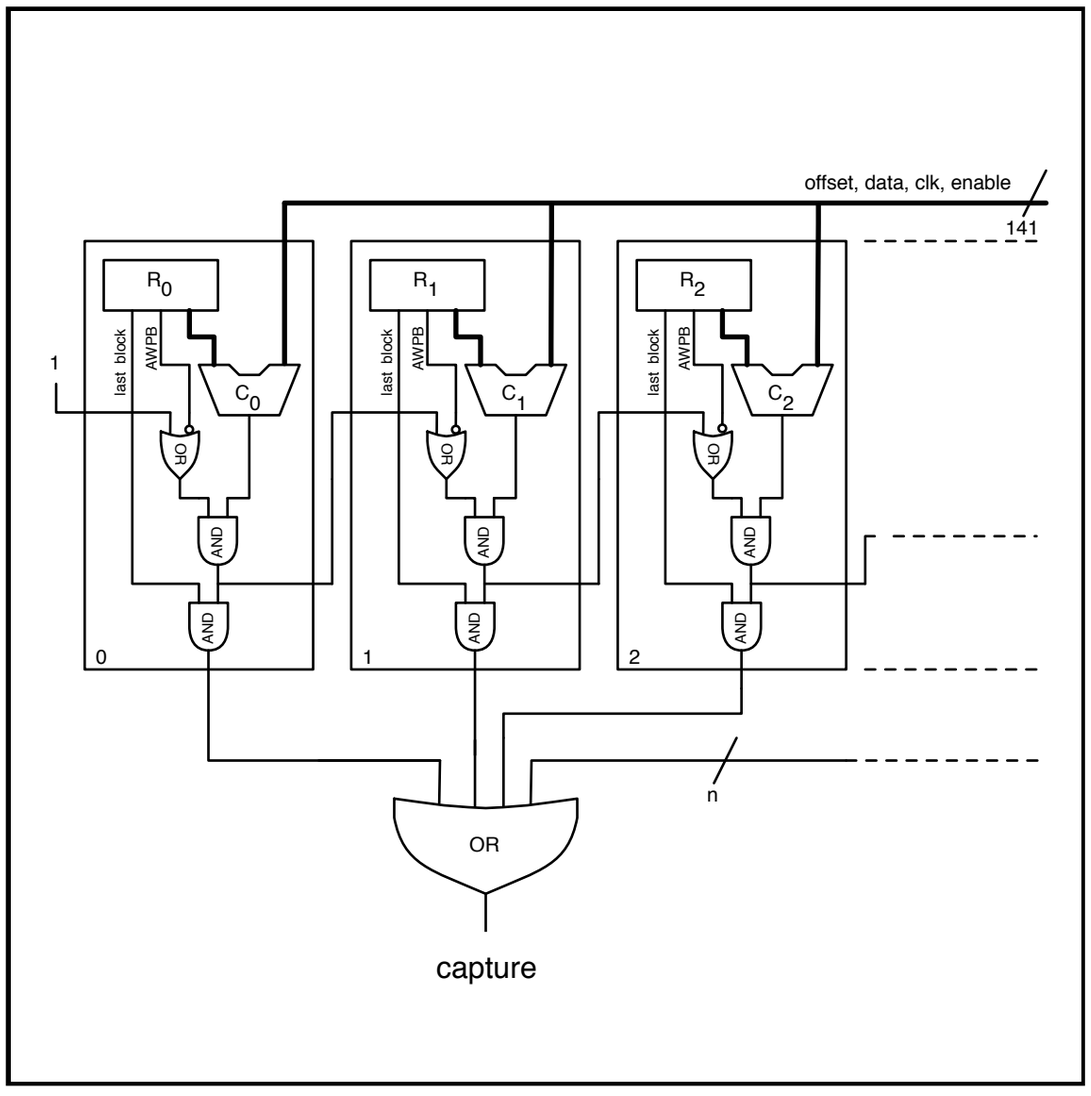
Details



Instr.	meaning
0.	Full 16 bit match.
1.	Full 16 bit not match. The outcome is true if the data does not match the pattern.
2.	Partial match, left half of pattern field contains the data to match, right half contains bit mask.
3.	Partial not match, left half of pattern field contains data to match, right half contains bitmask.
4.	Partial match, same as instruction 2 but left and right are re-versed.
5.	Partial not match, same as instruction 4 but left and right are reversed.
6.	Reserved for future use.
7.	Reserved for future use.

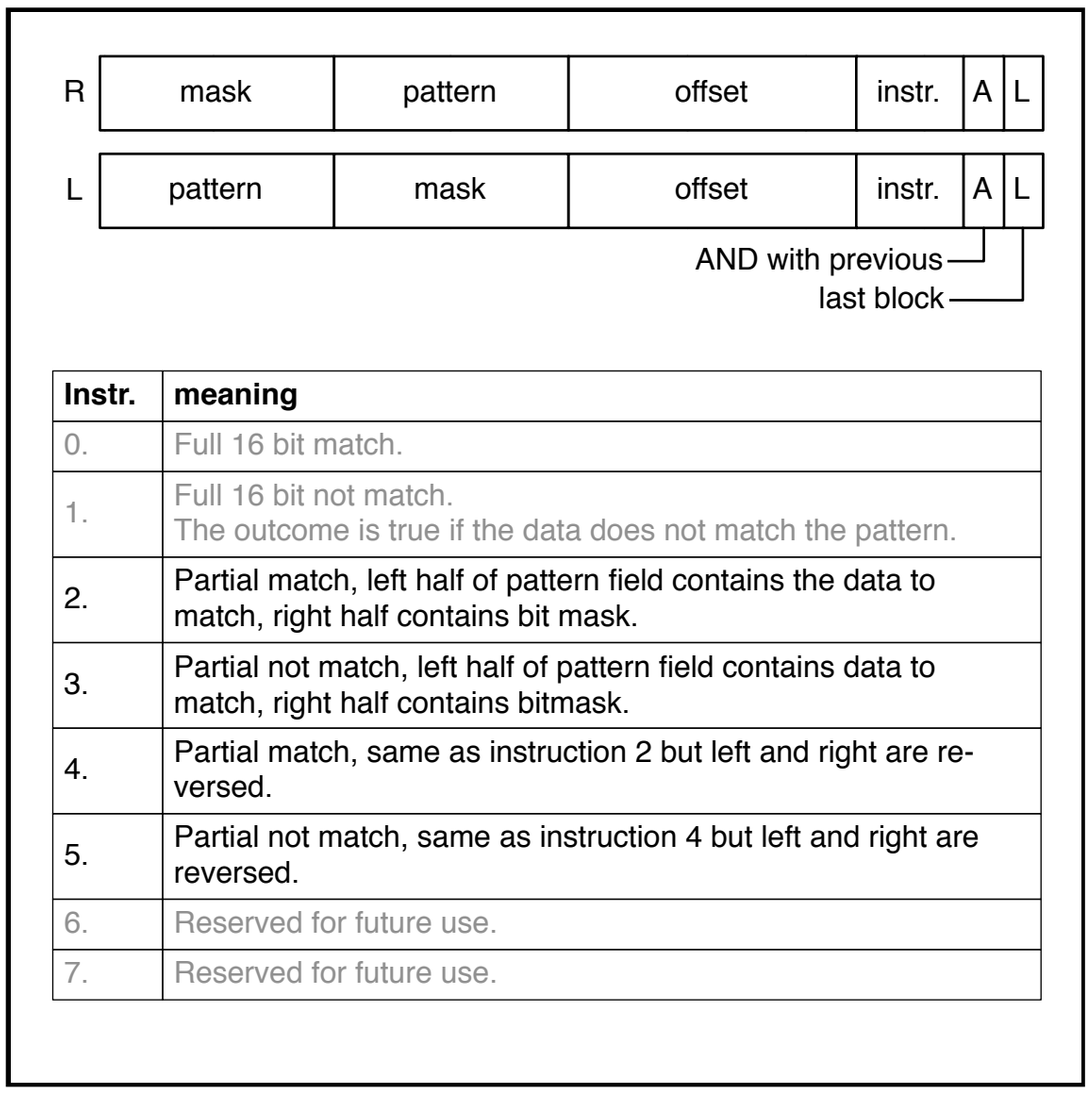
- Registers
  - Contain the pattern to match (16 bits)
  - The offset (11 bits)
    - 2048 bytes
    - 1522 byte frames
- Instruction (3 bits)
- Last two bits to handle products that do not one processor.

Details



- Registers
  - Contain the pattern to match (16 bits)
  - The offset (11 bits)
    - 2048 bytes
    - 1522 byte frames
- Instruction (3 bits)
- Last two bits to handle products that do not one processor.

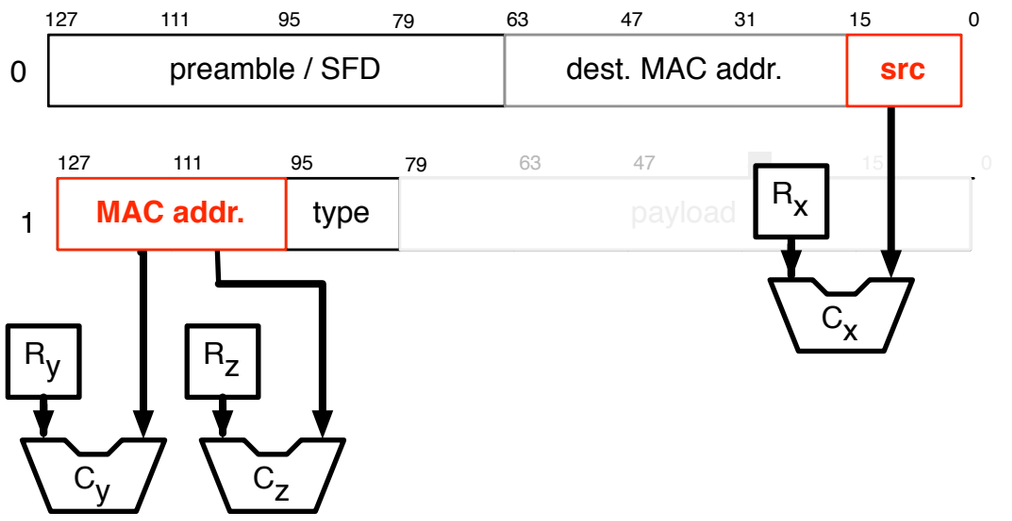
Details



Instr.	meaning
0.	Full 16 bit match.
1.	Full 16 bit not match. The outcome is true if the data does not match the pattern.
2.	Partial match, left half of pattern field contains the data to match, right half contains bit mask.
3.	Partial not match, left half of pattern field contains data to match, right half contains bitmask.
4.	Partial match, same as instruction 2 but left and right are re-versed.
5.	Partial not match, same as instruction 4 but left and right are reversed.
6.	Reserved for future use.
7.	Reserved for future use.

- Registers
  - Contain the pattern to match (16 bits)
  - The offset (11 bits)
    - 2048 bytes
    - 1522 byte frames
- Instruction (3 bits)
- Last two bits to handle products that do not one processor.
- Partial matches
  - Via a bit mask.
  - Per 8 bit patterns.

Details



not(ether src 00:01:02:03:04:05**)**

(R_x) 000101c5	00	01	e	1	0	1
(R_y) 02030205	02	03	10	1	0	1
(R_z) 04050245	04	05	12	1	0	1

input

not ether src 00:01:02:03:04:05

lexer/parser

!(N00e0001&N0100203&N0120405);

Quine-McCluskey

!N00e0001|!N0100203|!N0120405

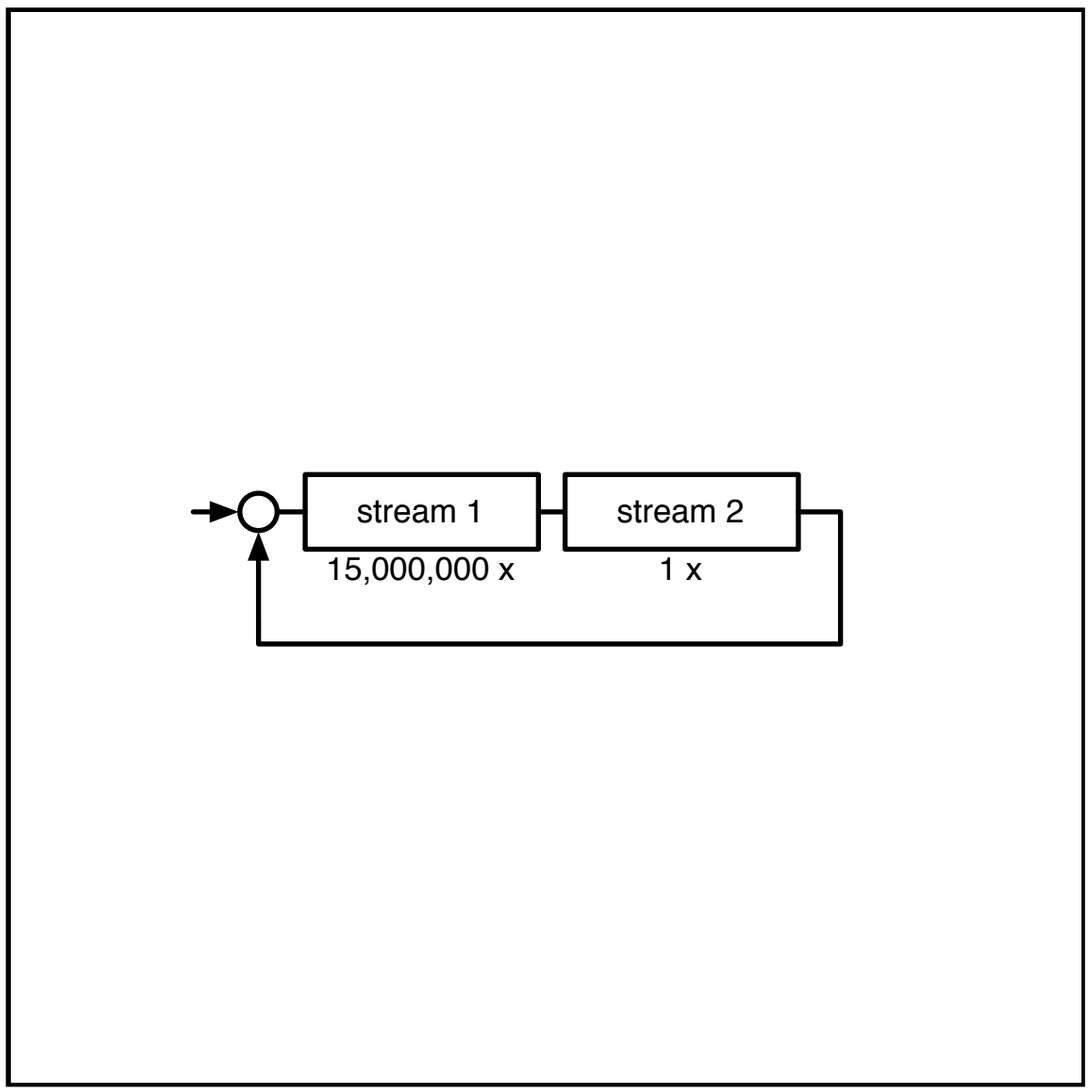
register mapper

(0)000101c5
(1)02030205
(2)04050245

Agenda

- An example.
- The problems with passive 10GbE monitoring.
  - Performance of generic computer hardware.
  - Issues with the mirror port feature.
- Overview of our solution.
  - Hardware architecture.
  - Homegrown firmware and software.
- A bit more detail.
- More to demo.

Demo



- Stream 1
  - 15,000,000 frames.
  - Source MAC: 00:01:02:03:04:05.
- Stream 2
  - 1 frame from another source MAC.

Conclusion...

- System works.

Conclusion...

- System works.
  - It has been useful already...

... & future development

- System works.
  - It has been useful already, but it is still in development.
    - BPF to instructions for the MISD processor.
    - Integrate the system in libpcap.
    - Traffic generator.