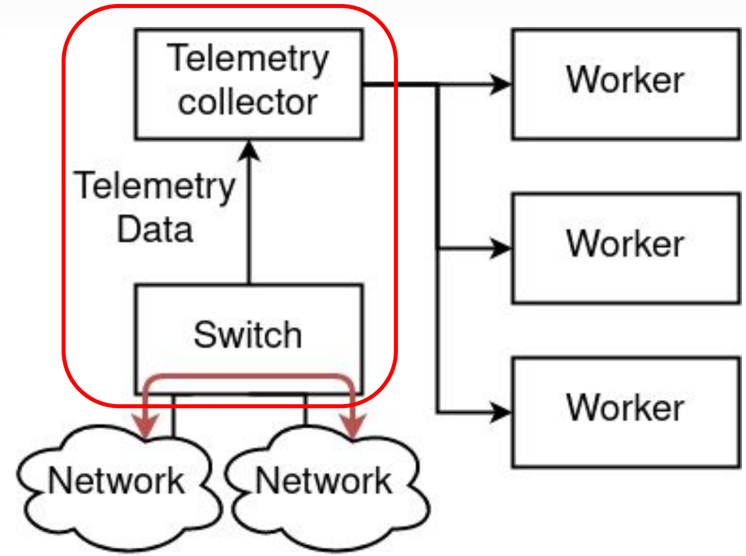
Collecting telemetry data using P4 and RDMA

Rutger Beltman
Silke Knossen

Supervisors:
Joseph Hill M.Sc.
Dr. Paola Grosso

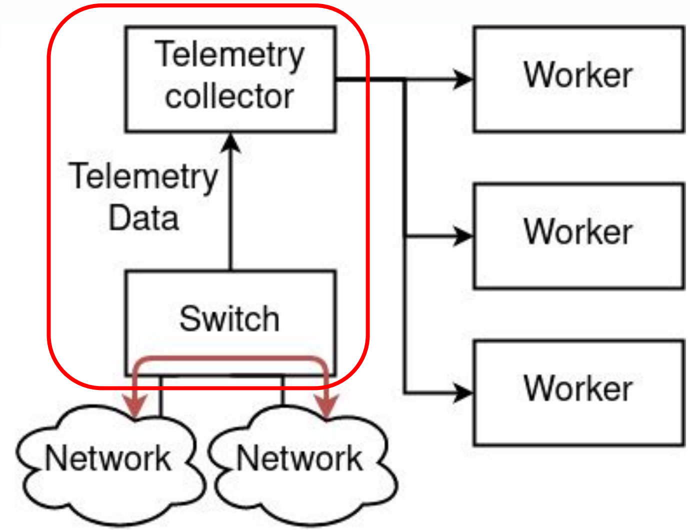
Introduction: Network Telemetry (I)

- Monitoring network health
- In-band network telemetry includes telemetry data in packets
- Delegate analyzation to multiple workers



Introduction: Network Telemetry (II)

- Requires an efficient means for collecting data
- Programming Protocol-independent Packet Processors (P4) for efficient telemetry data extraction
- Remote Direct Memory Access (RDMA) for efficient storage



Research Questions

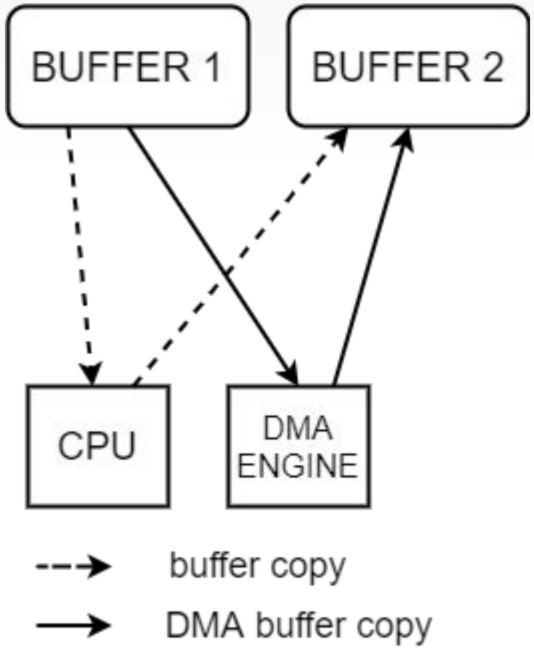
Can RDMA combined with P4 be used to efficiently collect telemetry data?

- How do we encapsulate telemetry data in an RDMA message?
- Can an RDMA session be maintained on a P4 switch?
- How can telemetry data be placed into persistent storage using RDMA?

DMA

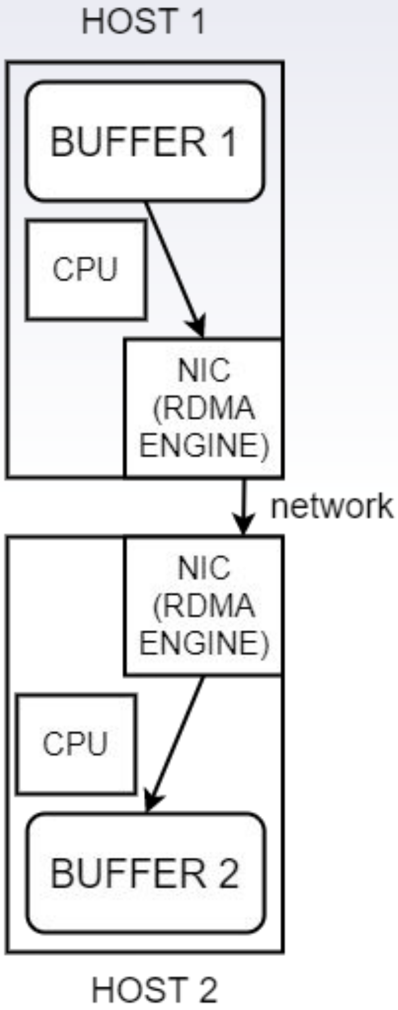
- Data is copied from buffer 1 to the buffer 2 via the CPU
- CPU spends a lot of cycles copying data
- Delagate high throughput transfers to DMA engine
- CPU can continue on other tasks while the DMA

  engine takes care of the transfer



RDMA

- Takes concept of DMA and puts it in the NIC
- Allows NIC to access data directly in memory
- CPU sets up a write operation
- The NIC on host 1 reads the buffer from memory and transfers it to the other NIC
- The NIC of host 2 writes the data to buffer 2
- The CPU is bypassed for the transfer of data



RoCEv1

- RDMA over Converged Ethernet version 1 (RoCEv1)
- RoCEv1 enables RDMA over layer 2 networks
- GRH has the same fields as IPv6
- BTH defines the RDMA operation for the NIC
- RETH includes memory address information for RDMA operations
- Invariant CRC is similar to Ethernet CRC, but slightly different

Related Work (I)

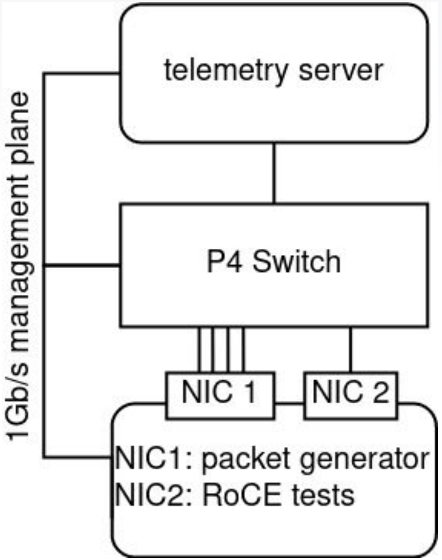
- Research by Tierney et al. (2012) compared the performance of TCP, UDP, UDT, and RoCE
  - CPU usage in RoCE is much less in comparison to the other protocols
  - RoCE showed consistently good performance
  - This research shows the potential of RoCE traffic in high-throughput networks

Related Work (II)

- Research by Kim et al. (2018) examined feasibility of implementing RoCE in P4 switch
  - Extending switch's buffer by storing burst data remotely
  - Extending forwarding tables by storing packet and action
  - Remotely increase counters for telemetry data
- "Borrowing" memory from remote server
- In our approach the server will eventually process this data further into the telemetry pipeline

Methodology & Setup

- Extract telemetry data with P4
- Implementing RoCE in P4 switch
- Send RoCE packet (RDMA write-only) with telemetry in payload
- Store payload on telemetry server

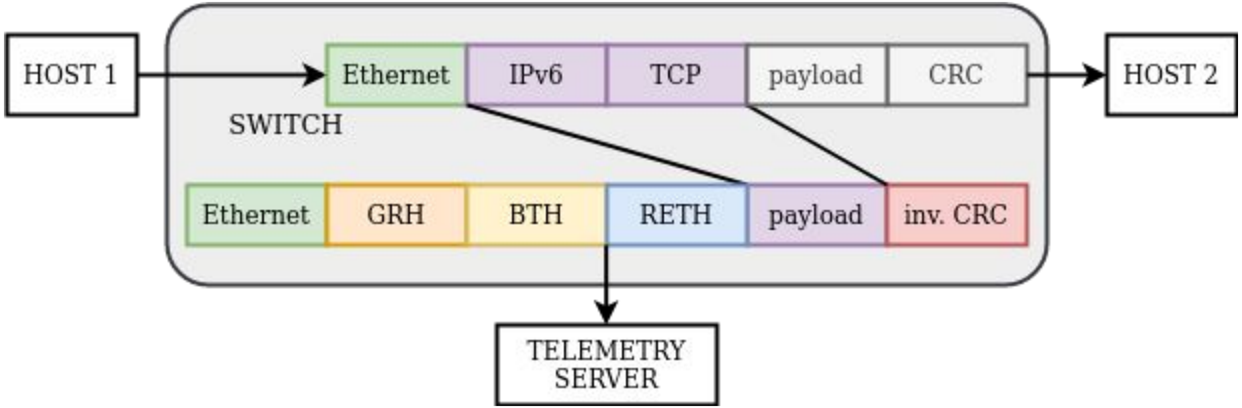


Server implementation

- Server uses *mmap* function to map virtual memory to a file on disk
- Set up the NIC to allow RDMA operations to the virtual memory address
- RDMA write-only can write directly to virtual memory, bypassing the CPU
- Open TCP socket to switch and share parameters required for RoCE packets

Switch implementation

- As there is no native support for RoCE on the switch, we create the RoCE headers from scratch in P4
- We learned the field values from the specification and experimentation



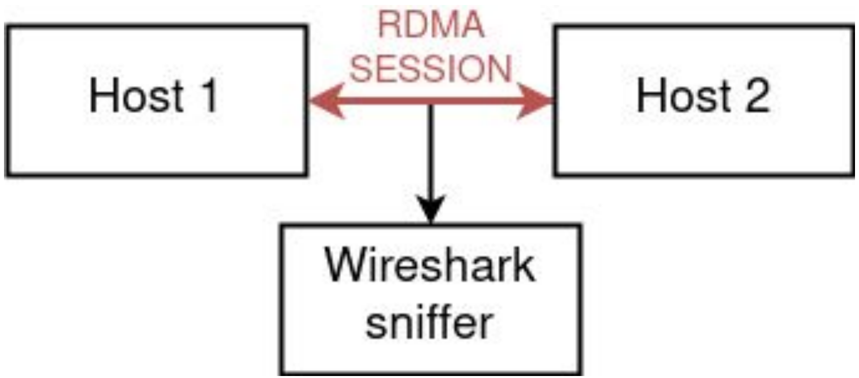
Switch: specific values

- Most of the header field values are static
- Others are dynamic or based on the server's RDMA parameters
  - Sequence number: counter increases with each packet
  - RDMA parameters from server are stored in a forwarding table
    - When the packet's egress port is to the telemetry server,
    - there is a match in the table
    - and the parameters are assigned to the packet
  - The virtual memory address is increased using an offset
  - CRC is calculated using an external function of the switch

Experiments (I)

Experiment 1: RoCEv1 experimentation to examine headers

- Establishing RDMA session between the two servers using RoCE libraries
- Analyze parameters that are used in the application and compare them to network traffic



Results experiment 1

```
▾ InfiniBand
  ▾ Global Route Header
      0110 .... = IP Version: 6
      .... 0000 0000 .... = Traffic Class: 0
      .... .... .... 0000 0000 0000 0000 0000 = Flow Label: 0
      Payload Length: 56
      Next Header: 27
      Hop Limit: 64
      Source GID: ::ffff:10.1.2.1
      Destination GID: ::ffff:10.1.2.2
  ▾ Base Transport Header
      Opcode: Reliable Connection (RC) - RDMA WRITE Only (10)
      0... .... = Solicited Event: False
      .1.. .... = MigReq: True
      ..11 .... = Pad Count: 3
      .... 0000 = Header Version: 0
      Partition Key: 65535
      Reserved: 00
      Destination Queue Pair: 0x000932
      1... .... = Acknowledge Request: True
      .000 0000 = Reserved (7 bits): 0
      Packet Sequence Number: 1
  ▾ RETH - RDMA Extended Transport Header
      Virtual Address: 26600384
      Remote Key: 178013
      DMA Length: 21
    Invariant CRC: 0xdd492f73
▾ Data (24 bytes)
    Data: 52444d41207772697465206f7065726174696f6e00000000
    [Length: 24]
```

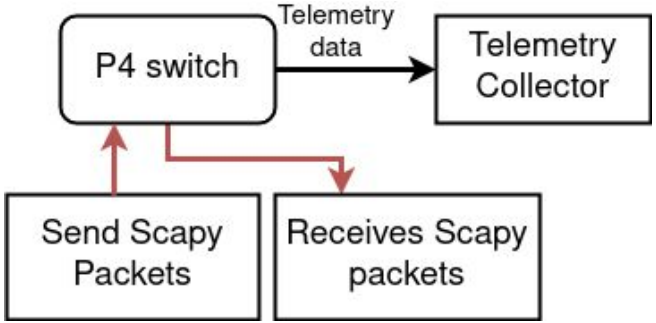
```
[rutger@sne-dtn-04 rdma-writeonly]$ ./rdma-
    tutorial -d mlx5_1 -i 1 -g 2
...
TCP connection was established
...
MR was registered with addr=0x195e3c0,
    lkey=0x2b75d, rkey=0x2b75d, flags=0x7
QP was created, QP number=0x932
...
completion was found in CQ with status 0x0
Contents of server buffer: 'RDMA write
    operation'

test result is 0
```

Experiments (II)

Experiment 2: RoCEv1 switch implementation testing

- Sending TCP packets crafted by Scapy from the Dell server
- Analyzed the file on the server to analyze correctness of the implementation




```
sendp(Ether()/IPv6(
    dst="fc00::1111:2222:3333:4444",
    src="fc00::5555:6666:7777:8888")/
        TCP(dport=111,sport=222,
            seq=0x1212,ack=0x3434),
    iface="rename5")
sendp(Ether()/IPv6(
    dst="fc00::5555:6666:7777:8888",
    src="fc00::1111:2222:3333:4444")/
        TCP(dport=222,sport=111,
            seq=0x3435,ack=0x1213),
    iface="rename5")
sendp(Ether()/IPv6(
    dst="fc00::1111:2222:3333:4444",
    src="fc00::5555:6666:7777:8888")/
        TCP(dport=111,sport=222,
            seq=0x1214,ack=0x3436),
    iface="rename5")
```

Results experiment 2

```
Supermicro $ hexdump -C /mnt/nvme/output5
00  |  fc00 0000 0000 0000  5555 6666 7777 8888
10  |  fc00 0000 0000 0000  1111 2222 3333 4444
20  |  00de 006f 0000 1212  0000 3434 0000 0000
30  |  fc00 0000 0000 0000  1111 2222 3333 4444
40  |  fc00 0000 0000 0000  5555 6666 7777 8888
50  |  006f 00de 0000 3435  0000 1213 0000 0001
60  |  fc00 0000 0000 0000  5555 6666 7777 8888
70  |  fc00 0000 0000 0000  1111 2222 3333 4444
80  |  00de 006f 0000 1214  0000 3436 0000 0002
```

```
sendp(Ether()/IPv6(
    dst="fc00::1111:2222:3333:4444",
    src="fc00::5555:6666:7777:8888")/
        TCP(dport=111,sport=222,
            seq=0x1212,ack=0x3434),
    iface="rename5")
sendp(Ether()/IPv6(
    dst="fc00::5555:6666:7777:8888",
    src="fc00::1111:2222:3333:4444")/
        TCP(dport=222,sport=111,
            seq=0x3435,ack=0x1213),
    iface="rename5")
sendp(Ether()/IPv6(
    dst="fc00::1111:2222:3333:4444",
    src="fc00::5555:6666:7777:8888")/
        TCP(dport=111,sport=222,
            seq=0x1214,ack=0x3436),
    iface="rename5")
```

Discussion

- No CPU involvement means CPU does not know anything about the data
- No signalling: signalling should provide method to let the CPU know when data can be read from memory
- P4 has no support for packet trailers, limiting the payload length

Conclusion

- RDMA is a feasible solution to communicate telemetry data to a collector
- P4 allows the original header to be encapsulated into a RoCE packet
- An RDMA session is maintained on the switch by keeping state of required parameters
- *mmap* provides the possibility of mapping a file to virtual memory, allowing RDMA access to this memory region

Future work

- Comparing the performance of this implementation with other techniques
  - Data Plane Development Kit (DPDK)
  - extended Berkeley Packet Filter (eBPF)
- Optimizing system performance (NVMe over Fabric instead of memory mapping)
- Investigate in an efficient method to signal the CPU that data can be processed further into the telemetry pipeline
  - RDMA write-only with immediate
- Completing the telemetry pipeline by adding workers

Security implications

- Remote key is equivalent to a plain text password
- According to RFC 5040 manufacturers MUST ensure that only memory in a specific Protection Domain can be accessed.
- Full security considerations in RFC 5040 and RFC 5042
- Throwhammer is an RDMA variant on the Rowhammer attack
- If properly set up, security implications similar to UDP/TCP streams (traffic injection/sniffing).

CRC calculation

```
CRCPolynomial<bit<32>>(
        coeff = 0x04C11DB7,
        reversed = true,
        msb = false,
        extended = false,
        init = 0xFFFFFFFF,
        xor = 0xFFFFFFFF) poly;
```

Local Route Header	0xFFFFFFFF
Global Route Header	...
	Traffic Class: 0xFF
	Flow Label: 0xFFFFF
	...
	Hop Limit: 0xFF
	...
Base Transport Header	...
	Reserved: 0xFF
	...
RDMA Extended Transport Header	...
Payload	...

References

- (R)DMA figures inspired on:
  http://www.rdmaconsortium.org/home/The_Case_for_RDMA020531.pdf