Using P4-INT on Tofino for measuring device performance characteristics in a network lab

Speakers	Marcel Beausencourt (B.Eng), Max Julius Bode (B.Eng.)
Course of Studies	MA Information and Communication Engineering
	FB 1 – Energy and Information
	University of Applied Sciences Berlin (HTW Berlin)
	Prof. Dr. Thomas Scheffler

Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Agenda

1) Short introduction to P4

2) Inband Network Telemetry (INT)

3) Future prospects

What’s P4?

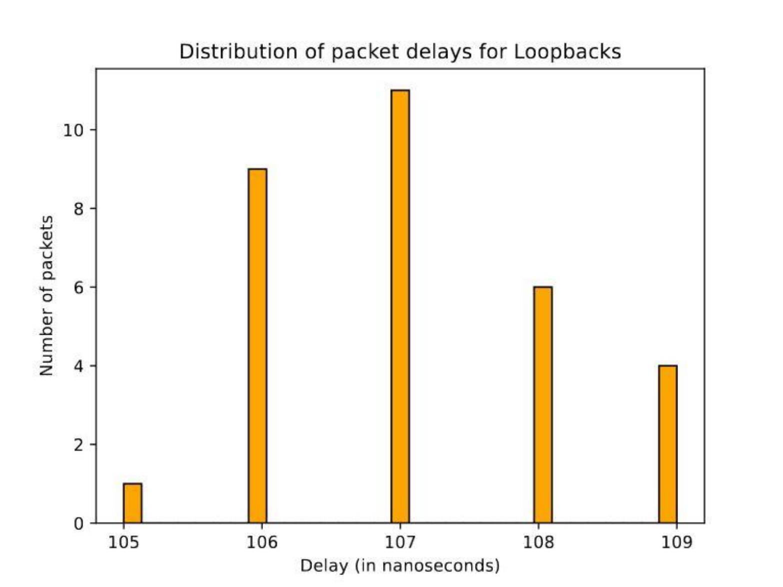
- Hardware-based programming of algorithms for packet processing
  - Create own packet processing behavior/algorithms
  - Manipulate every bit individually (→ headers)
- Successor of Open-Flow
- P4 Goals
  1. Reconfigurability
  2. Protocol independence
  3. Portability (architecture independence)

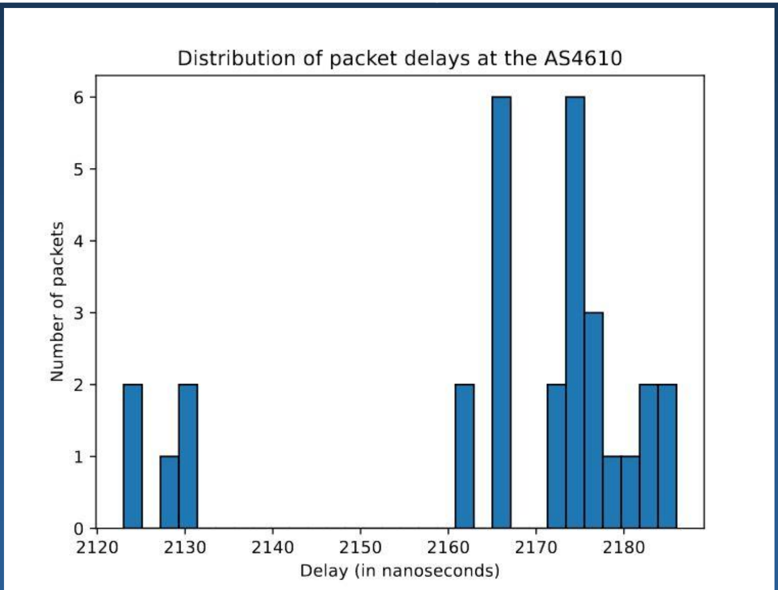
What can P4 do?

- Design and implementation of our own protocols
  - INT Header
  - Clone Header
- Pull unused protocols, optimize limited resources (TCAMs, …)
- Building networks for customers that are using proprietary protocols
- Research applications

First measurements with INT

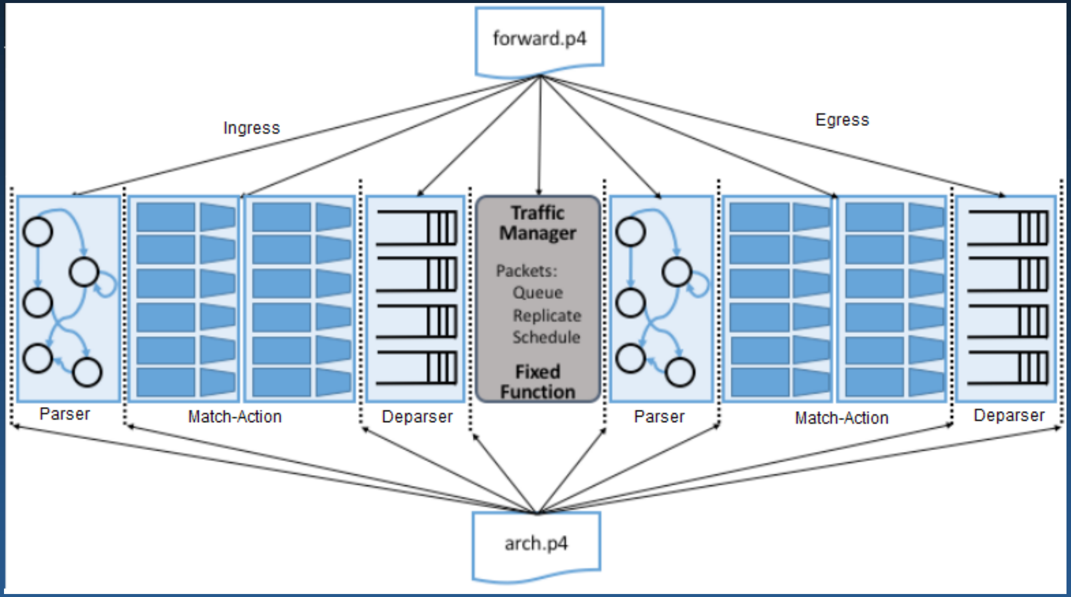
	Mean Delay in ns	Std. Dev. in ns
Loopback	106.96	1.06
AS4610	2128.96	16.58





How does P4 in Tofino work?

- Implement needed protocols
  - Parser (= State Machine)
- Implement your algorithms and tables
  - Match-Action
- Write the changes into the headers
  - Deparser (Checksums, MAC change, added headers ⟶ e.g. INT)
  - Send the packet ;)
- Payload isn‘t modified

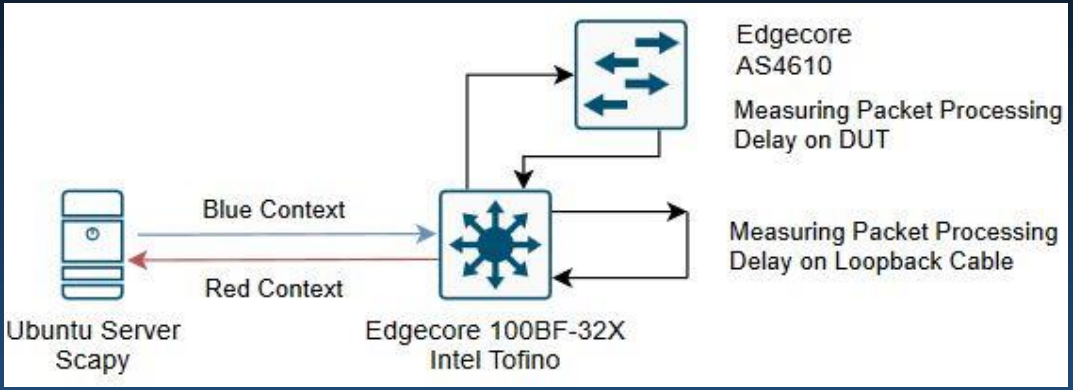


Original Source: https://sdn.systemsapproach.org/

INT Framework

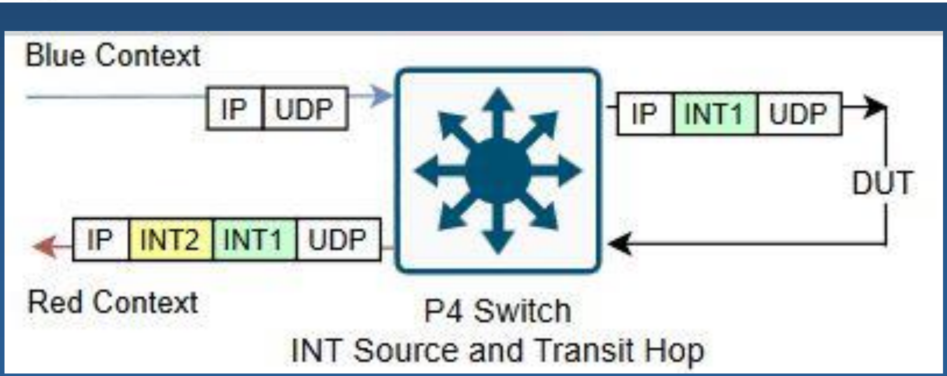
- Framework made by the P4-Language-Consrtium → Current version: 2.1
- Collect & report network state by the data plane (no interaction with the control plane is needed)
- INT-XD
  - No packet modifications
- INT-MX
  - Packets carry only instructions for the switches
- INT-MD
  - Supports instructions and metadata inside of network packets

Measuring delays with P4-INT



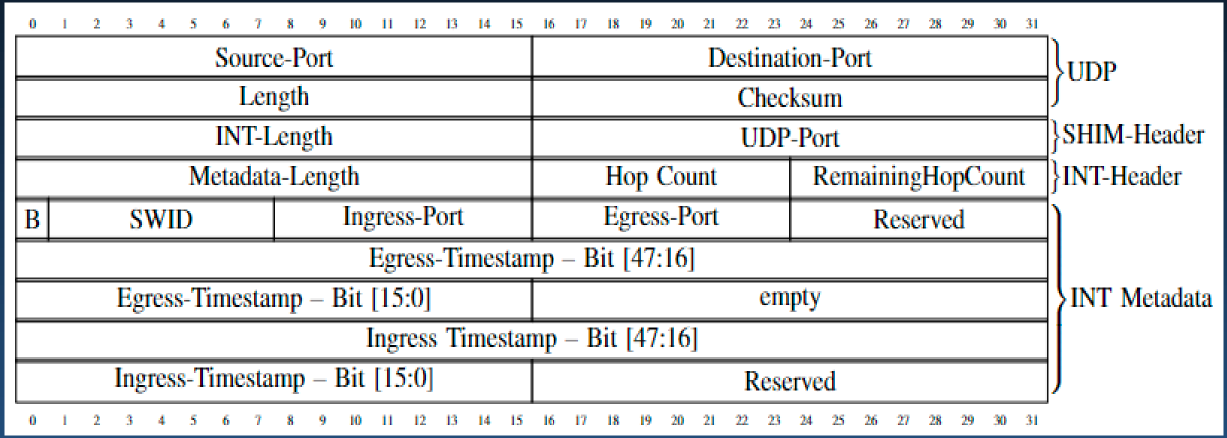
Edgecore
AS4610
Measuring Packet Processing Delay on DUT
Blue Context
Red Context
Measuring Packet Processing Delay on Loopback Cable
Ubuntu Server
Scapy
Edgecore 100BF-32X
Intel Tofino

Measuring topology



Adding INT headers into the packets

INT Header

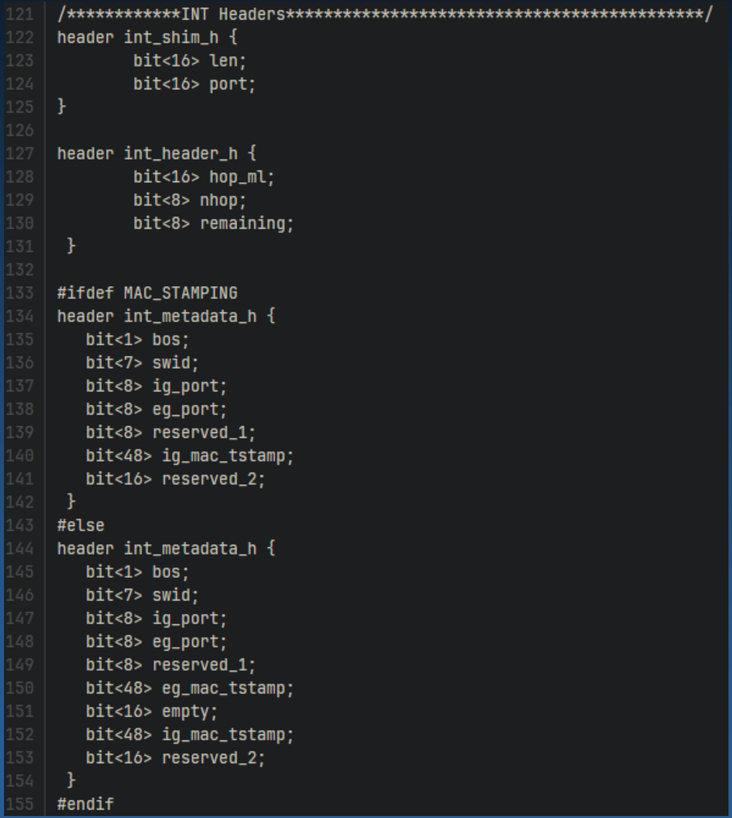


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Source-Port			Destination-Port		UDP
Length			Checksum		
INT-Length			UDP-Port		SHIM-Header
Metadata-Length			Hop Count	RemainingHopCount	INT-Header
B	SWID	Ingress-Port	Egress-Port	Reserved	INT Metadata
Egress-Timestamp – Bit [47:16]					
Egress-Timestamp – Bit [15:0]			empty		
Ingress Timestamp – Bit [47:16]					
Ingress-Timestamp – Bit [15:0]			Reserved		

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

INT Header format with 48-bit timestamps and P4 code implementation

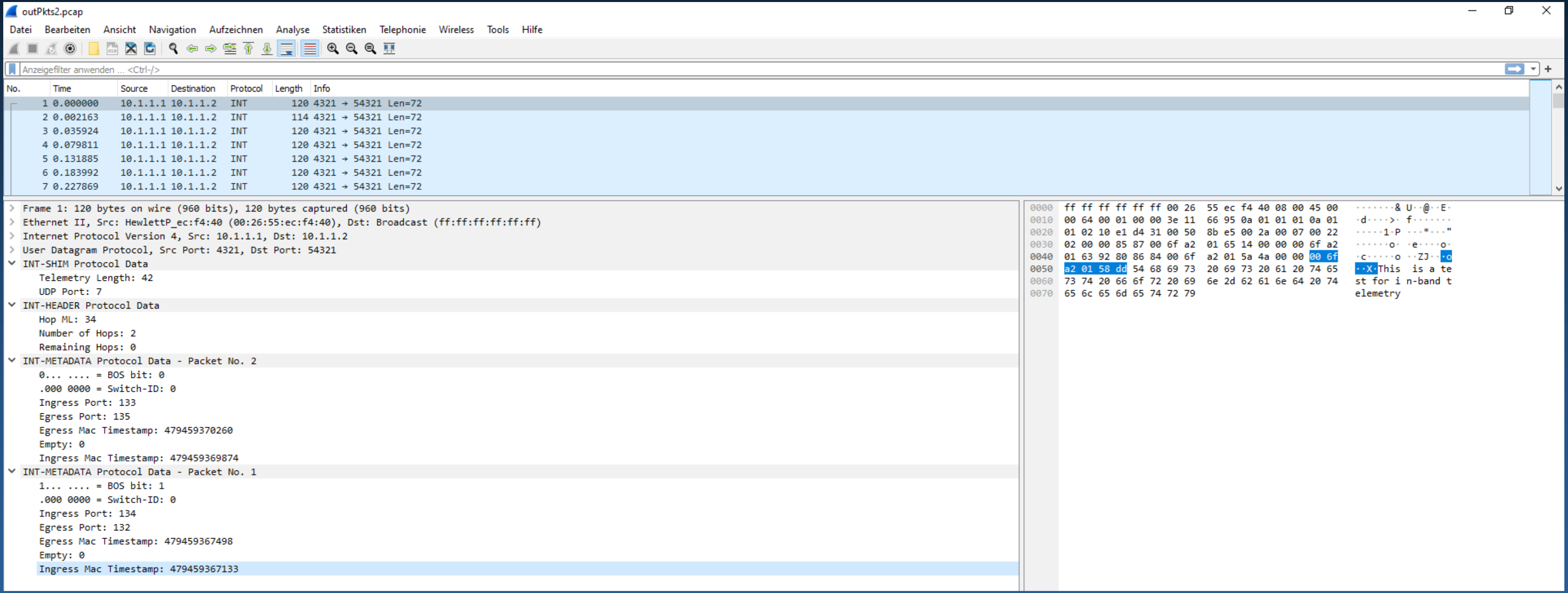


```
/************INT Headers***********************************************/
header int_shim_h {
        bit<16> len;
        bit<16> port;
}

header int_header_h {
        bit<16> hop_ml;
        bit<8> nhop;
        bit<8> remaining;
 }

#ifdef MAC_STAMPING
header int_metadata_h {
   bit<1> bos;
   bit<7> swid;
   bit<8> ig_port;
   bit<8> eg_port;
   bit<8> reserved_1;
   bit<48> ig_mac_tstamp;
   bit<16> reserved_2;
 }
#else
header int_metadata_h {
   bit<1> bos;
   bit<7> swid;
   bit<8> ig_port;
   bit<8> eg_port;
   bit<8> reserved_1;
   bit<48> eg_mac_tstamp;
   bit<16> empty;
   bit<48> ig_mac_tstamp;
   bit<16> reserved_2;
 }
#endif
```

What can be measured?

- Any networking device
  - Switches, Router
  - WDMs
- Effects/delays of different features (ACLs, Rate-Limits, ...)
- Edge Cases → Load-Testing of networks running very unusual protocol stacks

Traffic Analyzation I

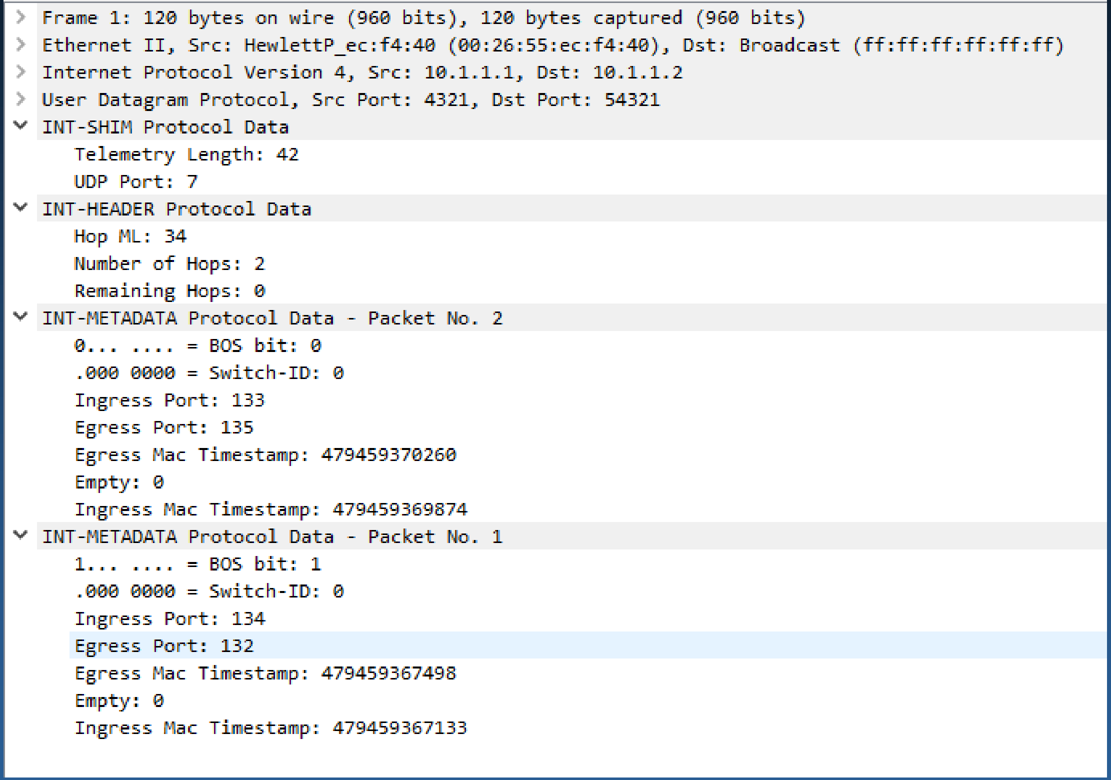


```
No.  Time      Source   Destination Protocol Length Info
   1 0.000000  10.1.1.1 10.1.1.2    INT         120 4321 → 54321 Len=72
   2 0.002163  10.1.1.1 10.1.1.2    INT         114 4321 → 54321 Len=72
   3 0.035924  10.1.1.1 10.1.1.2    INT         120 4321 → 54321 Len=72
   4 0.079811  10.1.1.1 10.1.1.2    INT         120 4321 → 54321 Len=72
   5 0.131885  10.1.1.1 10.1.1.2    INT         120 4321 → 54321 Len=72
   6 0.183992  10.1.1.1 10.1.1.2    INT         120 4321 → 54321 Len=72
   7 0.227869  10.1.1.1 10.1.1.2    INT         120 4321 → 54321 Len=72
```

```
> Frame 1: 120 bytes on wire (960 bits), 120 bytes captured (960 bits)
> Ethernet II, Src: HewlettP_ec:f4:40 (00:26:55:ec:f4:40), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.1.1.2
> User Datagram Protocol, Src Port: 4321, Dst Port: 54321
v INT-SHIM Protocol Data
     Telemetry Length: 42
     UDP Port: 7
v INT-HEADER Protocol Data
     Hop ML: 34
     Number of Hops: 2
     Remaining Hops: 0
v INT-METADATA Protocol Data - Packet No. 2
     0... .... = BOS bit: 0
     .000 0000 = Switch-ID: 0
     Ingress Port: 133
     Egress Port: 135
     Egress Mac Timestamp: 479459370260
     Empty: 0
     Ingress Mac Timestamp: 479459369874
v INT-METADATA Protocol Data - Packet No. 1
     1... .... = BOS bit: 1
     .000 0000 = Switch-ID: 0
     Ingress Port: 134
     Egress Port: 132
     Egress Mac Timestamp: 479459367498
     Empty: 0
     Ingress Mac Timestamp: 479459367133
```

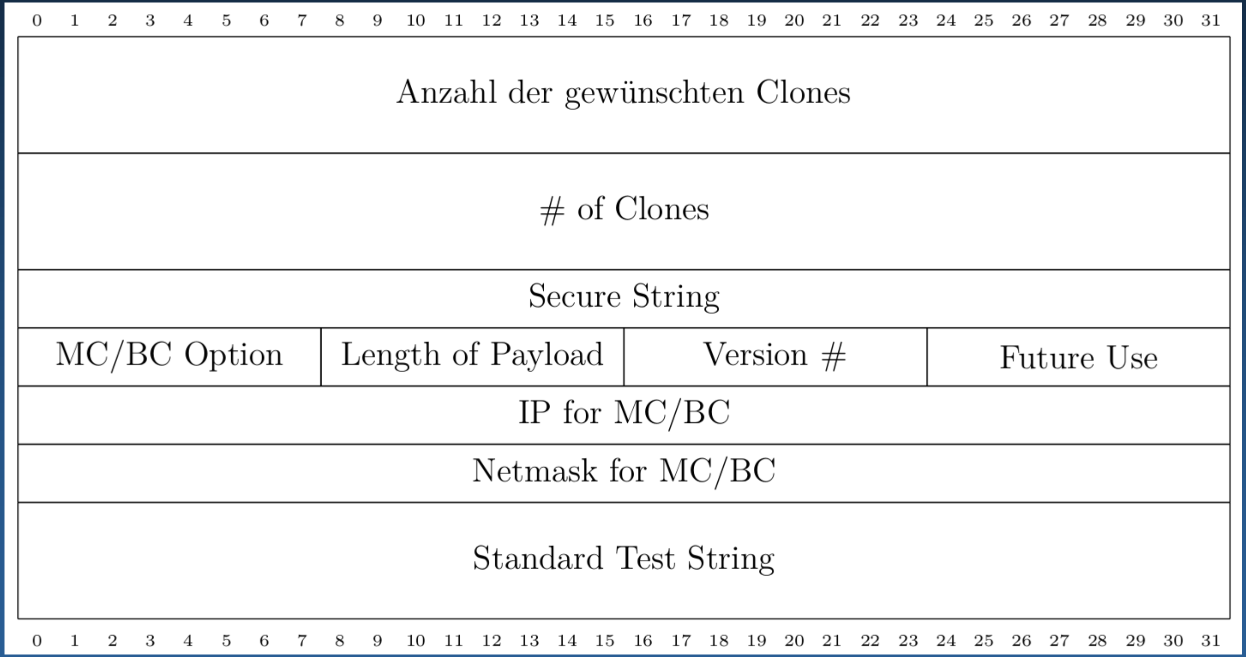
```
0000  ff ff ff ff ff ff 00 26  55 ec f4 40 08 00 45 00   ·······& U··@··E·
0010  00 64 00 01 00 00 3e 11  66 95 0a 01 01 01 0a 01   ·d····>· f·······
0020  01 02 10 e1 d4 31 00 50  8b e5 00 2a 00 07 00 22   ·····1·P ···*···"
0030  02 00 00 85 87 00 6f a2  01 65 14 00 00 00 6f a2   ······o· ·e····o·
0040  01 63 92 80 86 84 00 6f  a2 01 5a 4a 00 00 00 6f   ·c·····o ··ZJ···o
0050  a2 01 58 dd 54 68 69 73  20 69 73 20 61 20 74 65   ··X·This  is a te
0060  73 74 20 66 6f 72 20 69  6e 2d 62 61 6e 64 20 74   st for i n-band t
0070  65 6c 65 6d 65 74 72 79                            elemetry
```

Traffic Analyzation II



```
> Frame 1: 120 bytes on wire (960 bits), 120 bytes captured (960 bits)
> Ethernet II, Src: HewlettP_ec:f4:40 (00:26:55:ec:f4:40), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.1.1.2
> User Datagram Protocol, Src Port: 4321, Dst Port: 54321
v INT-SHIM Protocol Data
     Telemetry Length: 42
     UDP Port: 7
v INT-HEADER Protocol Data
     Hop ML: 34
     Number of Hops: 2
     Remaining Hops: 0
v INT-METADATA Protocol Data - Packet No. 2
     0... .... = BOS bit: 0
     .000 0000 = Switch-ID: 0
     Ingress Port: 133
     Egress Port: 135
     Egress Mac Timestamp: 479459370260
     Empty: 0
     Ingress Mac Timestamp: 479459369874
v INT-METADATA Protocol Data - Packet No. 1
     1... .... = BOS bit: 1
     .000 0000 = Switch-ID: 0
     Ingress Port: 134
     Egress Port: 132
     Egress Mac Timestamp: 479459367498
     Empty: 0
     Ingress Mac Timestamp: 479459367133
```

Packet-Duplication / Clone-Header



<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="32">Anzahl der gewünschten Clones</td></tr>
<tr><td colspan="32"># of Clones</td></tr>
<tr><td colspan="32">Secure String</td></tr>
<tr><td colspan="8">MC/BC Option</td><td colspan="8">Length of Payload</td><td colspan="8">Version #</td><td colspan="8">Future Use</td></tr>
<tr><td colspan="32">IP for MC/BC</td></tr>
<tr><td colspan="32">Netmask for MC/BC</td></tr>
<tr><td colspan="32">Standard Test String</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
</table>

- Makes X clones of a packet → 1   field
- Compares wanted number of packets with real number of packet → "# of Clones" = Counter
- Relies on Ethernet and IPv4 → Clone = Layer4
- Secure String → Authorization
- MC/BC Option → enables Multi-/Broadcast
  - In combination with IP and Netmask for MC/BC

Goals and future prospects

- Generation of flexible test-traffic >= 100 GBit/s
- Suited for analyzing and learning network protocols
  - Header structures and implementation
  - Basic algorithms
  - See what a network device has to do to forward a packet
- Front-End Application for INT, Scapy

Thank you for your attention!

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Contact

- Marcel Beausencourt
    - Marcel.beausencourt@student.htw-berlin.de
- Max Julius Bode
    - Max.bode@student.htw-berlin.de

Prof. Dr. Thomas Scheffler

- Thomas.scheffler@htw-berlin.de

- Git Repo incl. Bachelor thesis for getting an easier start into P4
    - https://github.com/Selltowitz/p4