A Hands-on Workshop on P4 Programmable Switches

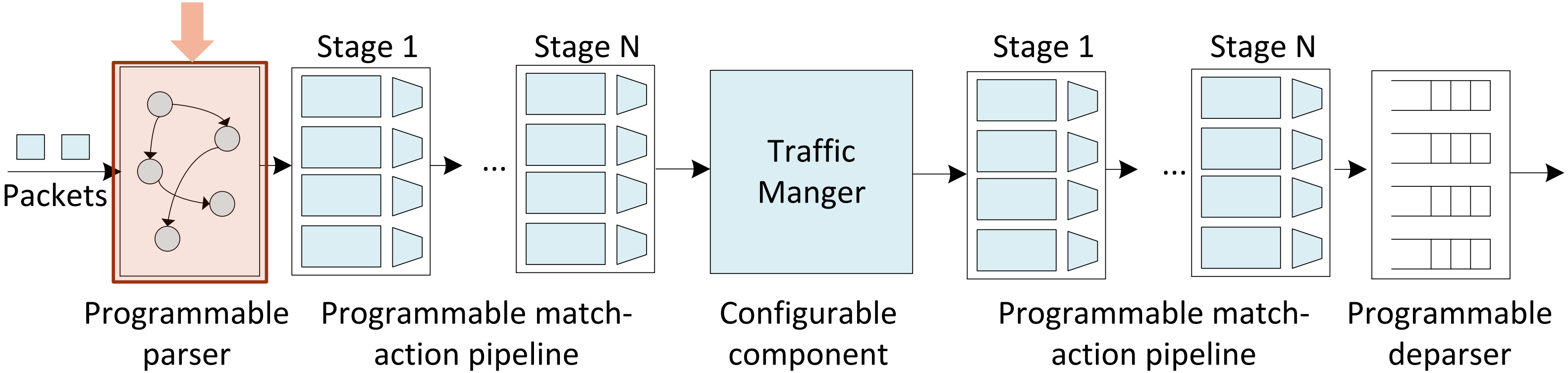
Elie Kfoury, Ali AlSabeh, Jose Gomez
University of South Carolina
http://ce.sc.edu/cyberinfra
ekfoury@email.sc.edu, aalsabeh@email.sc.edu, gomezgaj@email.sc.edu

February 16th, 23rd, 2022

Hands on Session 2: Writing a Parser for IPv4 and IPv6

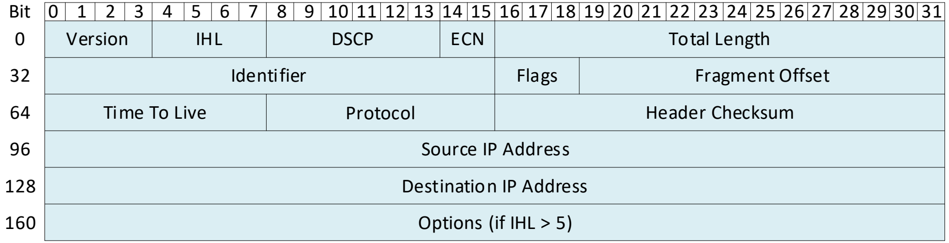
Programmable Parser

- The parser enables parsing arbitrary headers with a finite state machine
- The state machine defines the order of the headers within the packets
- The packet is split into the defined headers and the remaining is treated as the payload

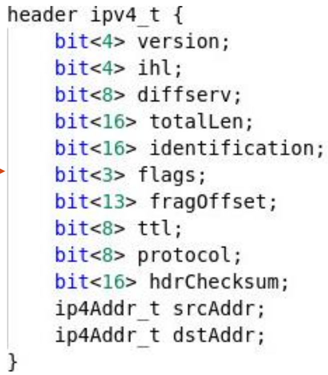


Packet Headers

- The packet headers are specified by the programmer
- The programmer has the flexibility of defining custom/non-standardized headers
- Such capability is not available in non-programmable devices



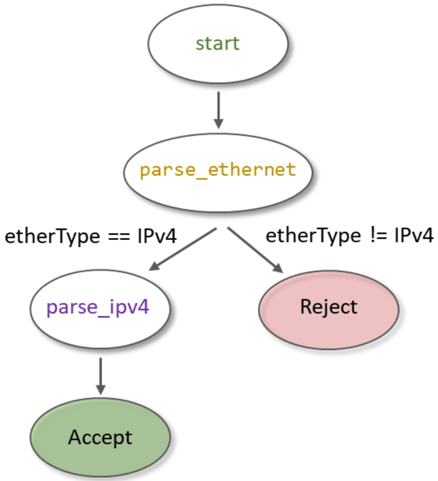
<table>
<tr><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>0</td><td colspan="4">Version</td><td colspan="4">IHL</td><td colspan="6">DSCP</td><td colspan="2">ECN</td><td colspan="16">Total Length</td></tr>
<tr><td>32</td><td colspan="16">Identifier</td><td colspan="3">Flags</td><td colspan="13">Fragment Offset</td></tr>
<tr><td>64</td><td colspan="8">Time To Live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><td>96</td><td colspan="32">Source IP Address</td></tr>
<tr><td>128</td><td colspan="32">Destination IP Address</td></tr>
<tr><td>160</td><td colspan="32">Options (if IHL &gt; 5)</td></tr>
</table>

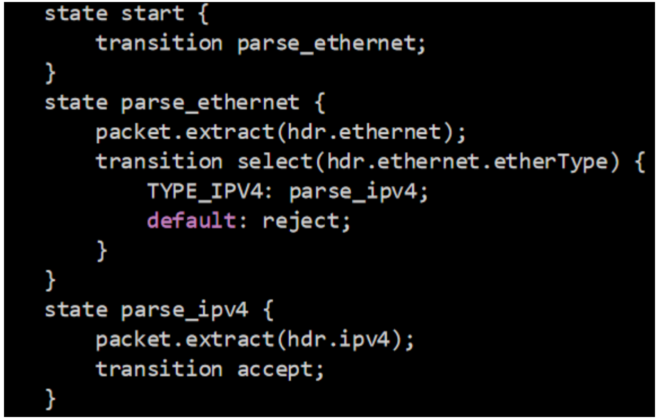


```
header ipv4_t {
    bit<4> version;
    bit<4> ihl;
    bit<8> diffserv;
    bit<16> totalLen;
    bit<16> identification;
    bit<3> flags;
    bit<13> fragOffset;
    bit<8> ttl;
    bit<8> protocol;
    bit<16> hdrChecksum;
    ip4Addr_t srcAddr;
    ip4Addr_t dstAddr;
}
```

Programmable Parser

- The parser enables declaring arbitrary headers with a finite state machine
- The state machine defines the order of the headers within the packets



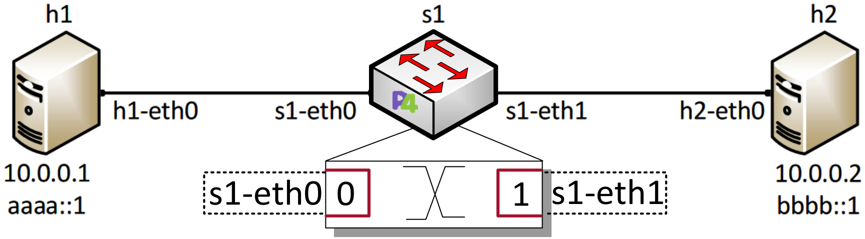


```
state start {
    transition parse_ethernet;
}
state parse_ethernet {
    packet.extract(hdr.ethernet);
    transition select(hdr.ethernet.etherType) {
        TYPE_IPV4: parse_ipv4;
        default: reject;
    }
}
state parse_ipv4 {
    packet.extract(hdr.ipv4);
    transition accept;
}
```

Lab 4: Parser Implementation

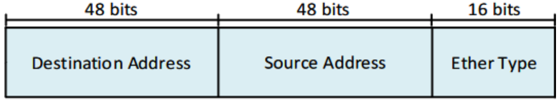
Lab Topology and Objectives

- The topology consists of two hosts: h1 and h2; one P4 switch: s1
- Defining the headers for Ethernet, IPv4 and IPv6
- Implementing the parser
- Testing and verifying the switch behavior when IPv4 and IPv6 packets are received



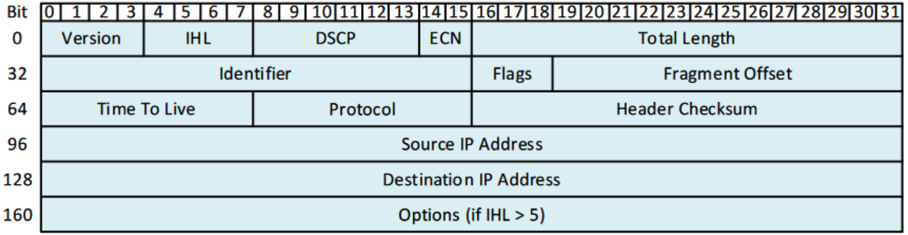
Headers Format

- Ethernet header:



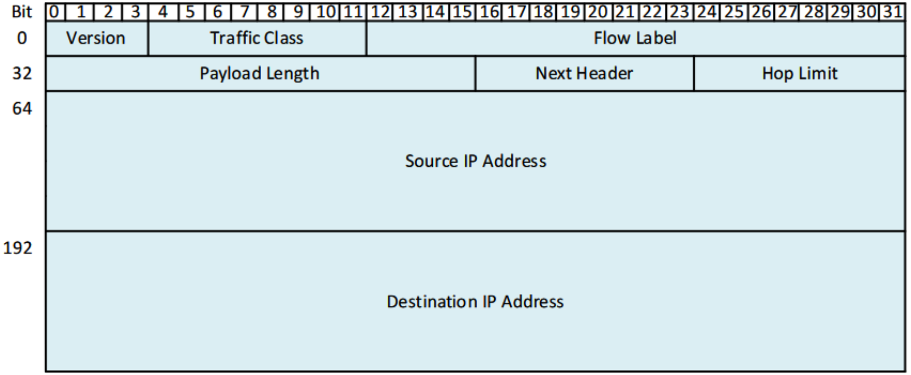
48 bits	48 bits	16 bits
Destination Address	Source Address	Ether Type

- IPv4 header:



Bit	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
0	Version				IHL				DSCP						ECN		Total Length															
32	Identifier																Flags			Fragment Offset												
64	Time To Live								Protocol								Header Checksum															
96	Source IP Address																															
128	Destination IP Address																															
160	Options (if IHL > 5)																															

- IPv6 header:



Bit	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
0	Version				Traffic Class								Flow Label																			
32	Payload Length																Next Header								Hop Limit							
64	Source IP Address																															
192	Destination IP Address																															