Abstracting switch architectures - the P4 approach -

Sept 10, 2015

Mihai Budiu

barefootnetworks.com

The P4 tension

Universal

Customizable

P4 v1:
Fixed Abstract Forwarding Model

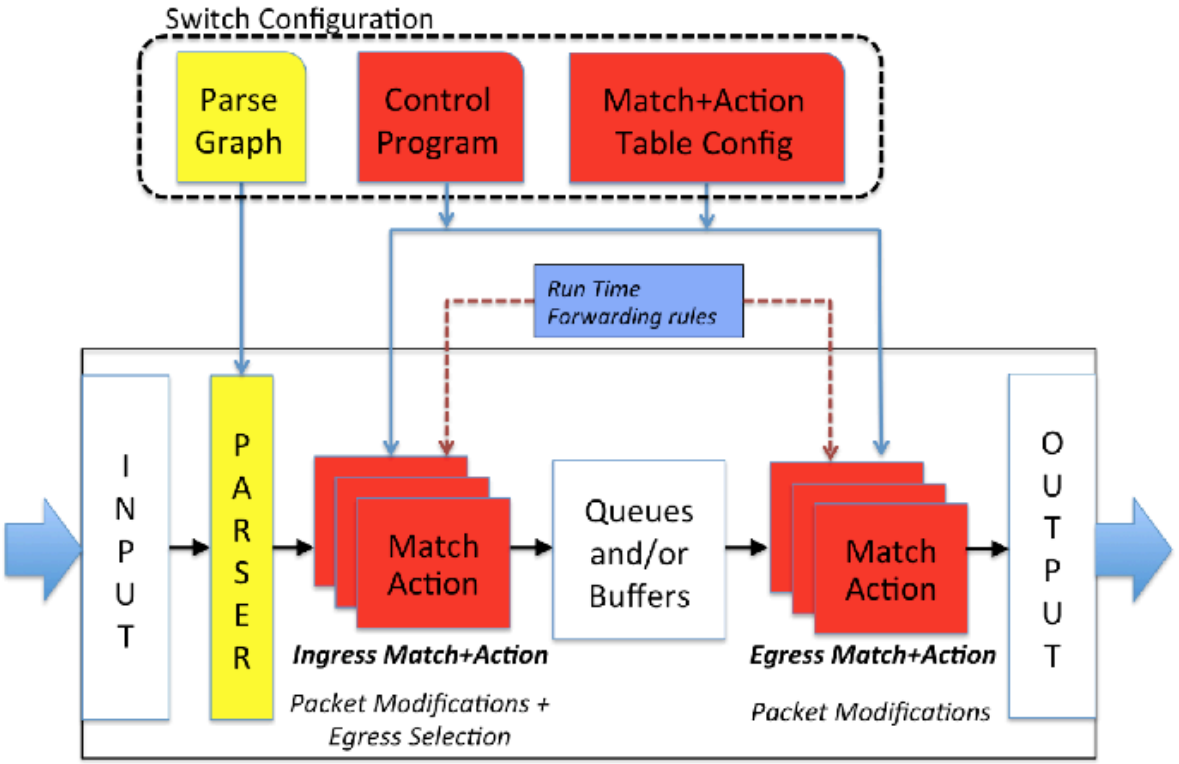


Figure 1: Abstract Forwarding Model

P4 v1: Details

Identifier	Exception Event
p4_pe_index_out_of_bounds	A header stack array in… cared bound.
p4_pe_out_of_packet	There were no… m- plete an…
p4_pe_header_too_long	A c… the declared
p4_pe_header_too_short	…was less than the min- …ixed length portion of the
p4_pe_unh…	…ment had no default specified but the …on value was not in the case list.
…	…ecksum error was detected.
	This is not an exception itself, but allows the pro- grammer to define a handler to specify the default behavior if no handler for the condition exists.

Parser exceptions

Table 2: Standard Parser Exceptions

Action profiles

Mirroring

field_list_calculation

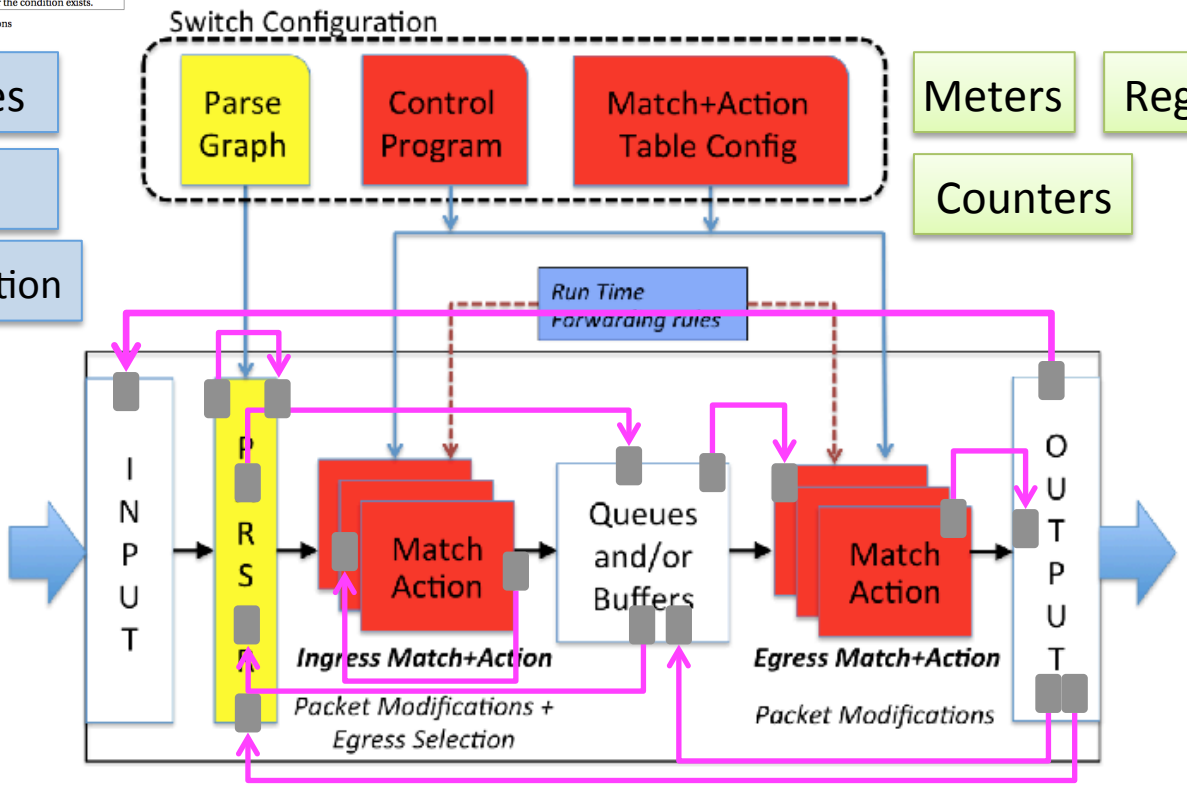
xor16

csum16

crc16

crc32

programmable

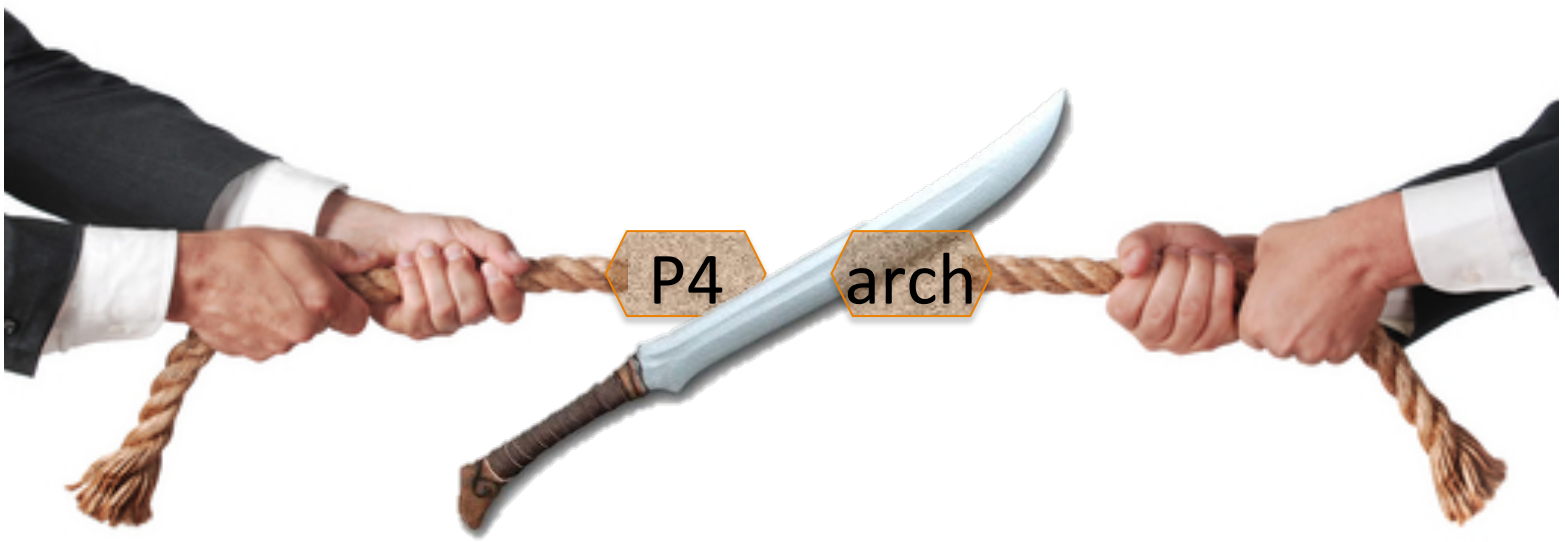


Field	Notes
ingress_port	The port on which the packet arrived. Set prior to parsing. Always defined. Read only
packet_length	The number of bytes in the par… Ethernet, does not include the CRC. Set … Can- not be used for matching or … if the switch is in "cut-throug…
egress_spec	Specification of an egres… match+action during … the "intended" egress… physical port(s) (ser… physical port, a lo… LAG, a route, or … group.
egress_port	The physica… committed… by the B… egress … Read
egress_instance	An …
instance_type	… of instance of the packet: … one … cated … flags or type value; for example, could a … recirculated? Do we need a counter too?
parser_…	… of the parser. 0 means no error. Otherwise, …alue indicates what error occurred during pars- …g. Specific representation is TBD.
pars…	…f a parser error occurred, this is an indication of the location in the parser program where the error oc- curred. Specific representation is TBD.

Intrinsic metadata

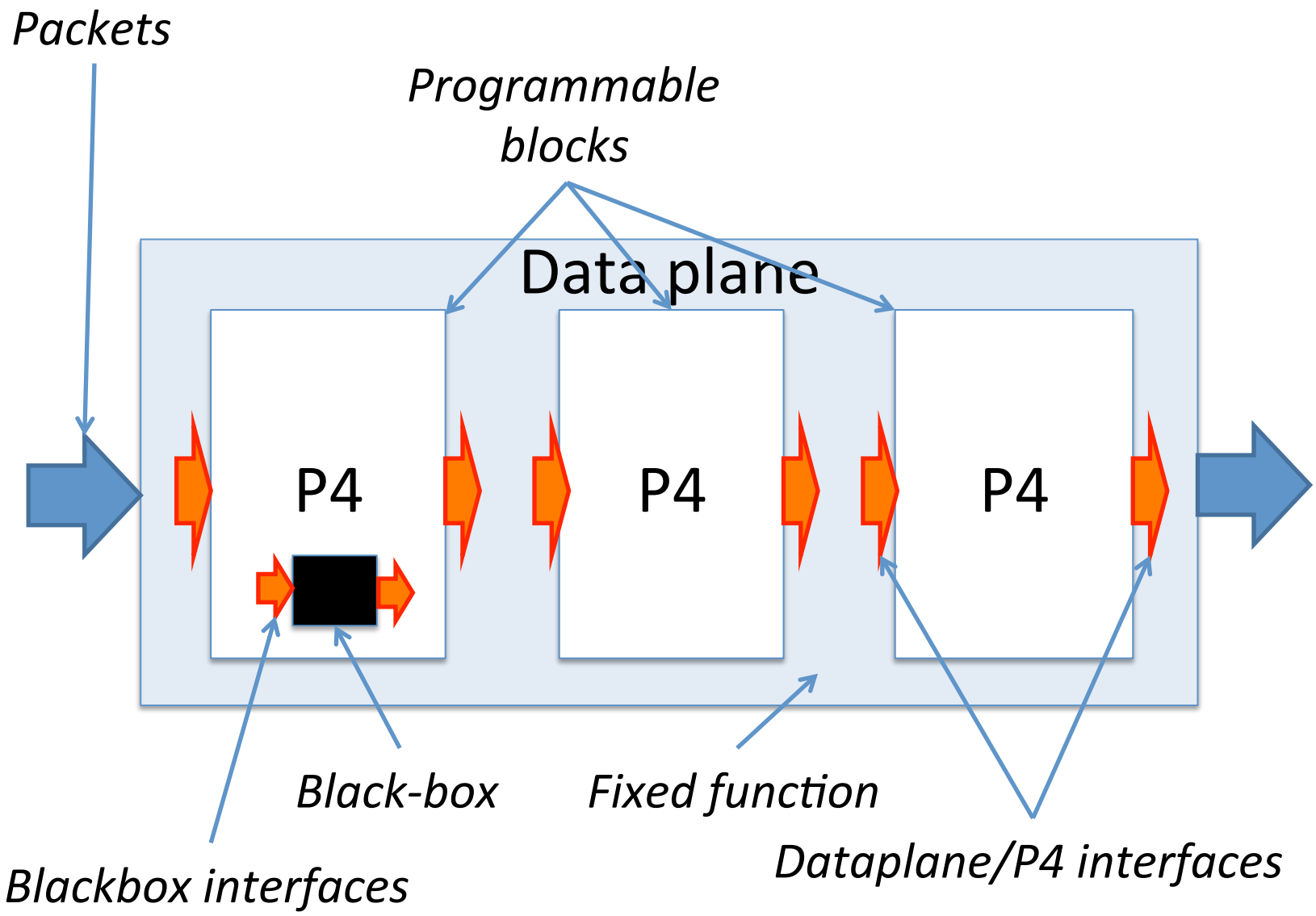
Table 3: Standard Intrinsic Metadata Fields

Divide and conquer

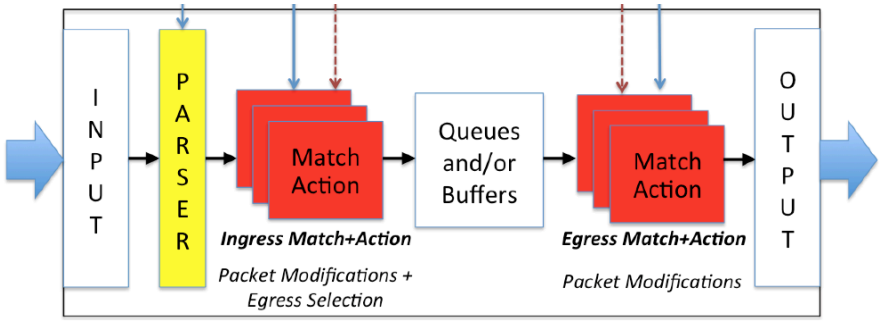


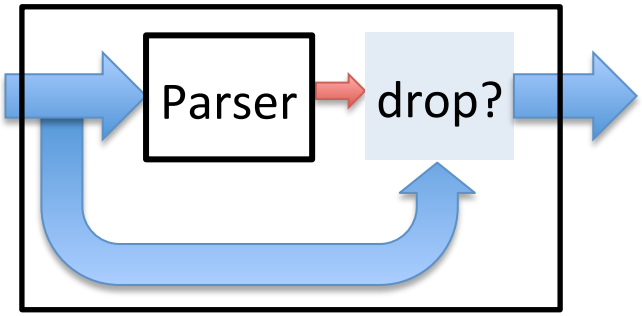
- Separate language definition from architecture definition
- Evolve them independently

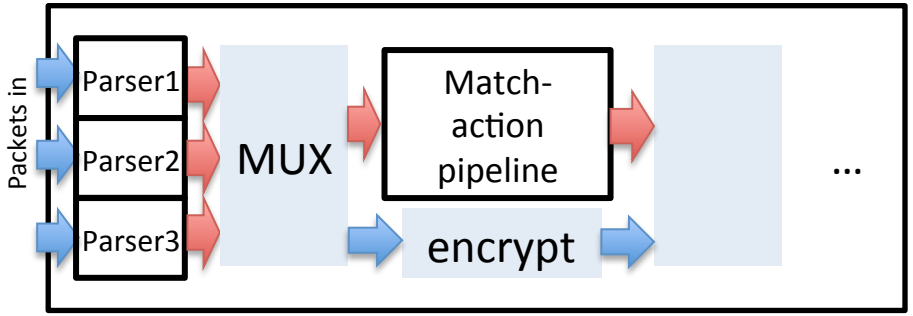
Generic Programmable Dataplane Model



P4 Support for multiple architectures

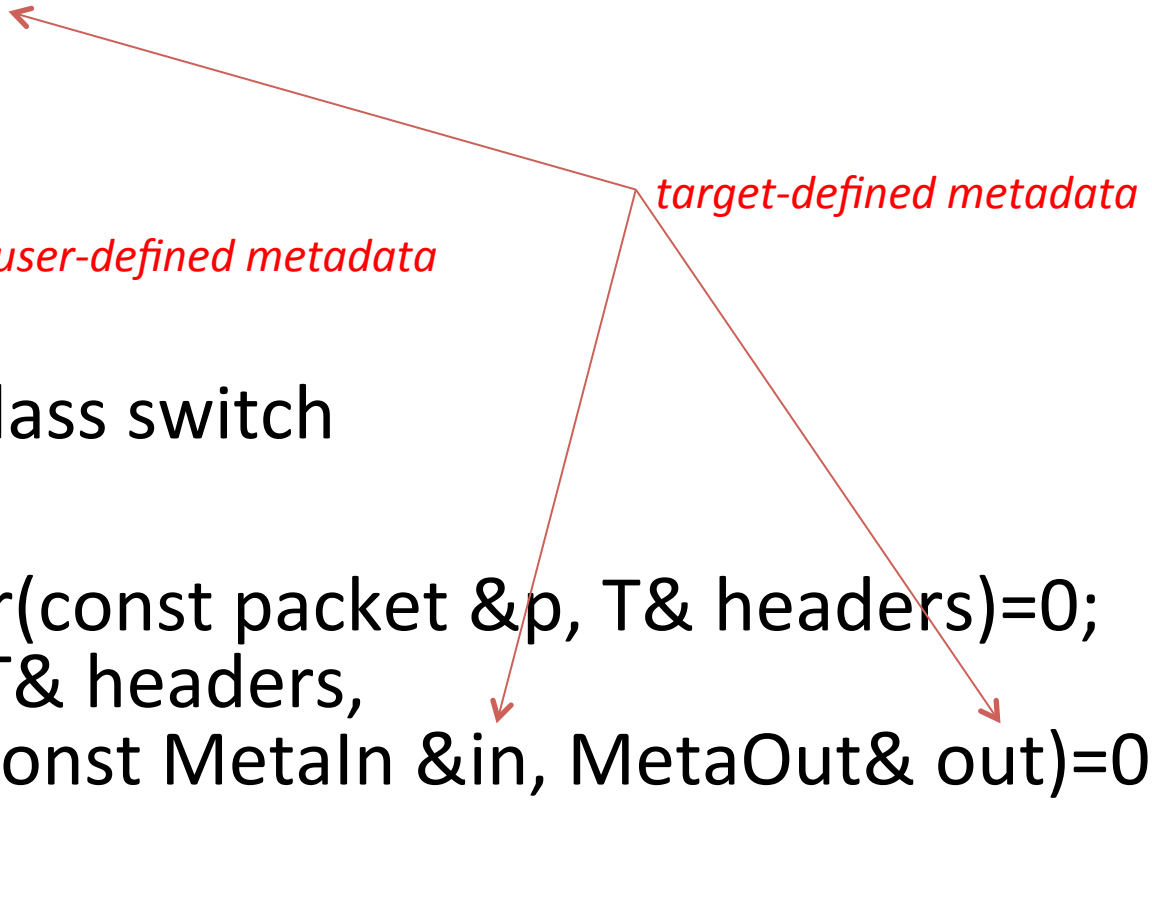






Switch architecture in C++

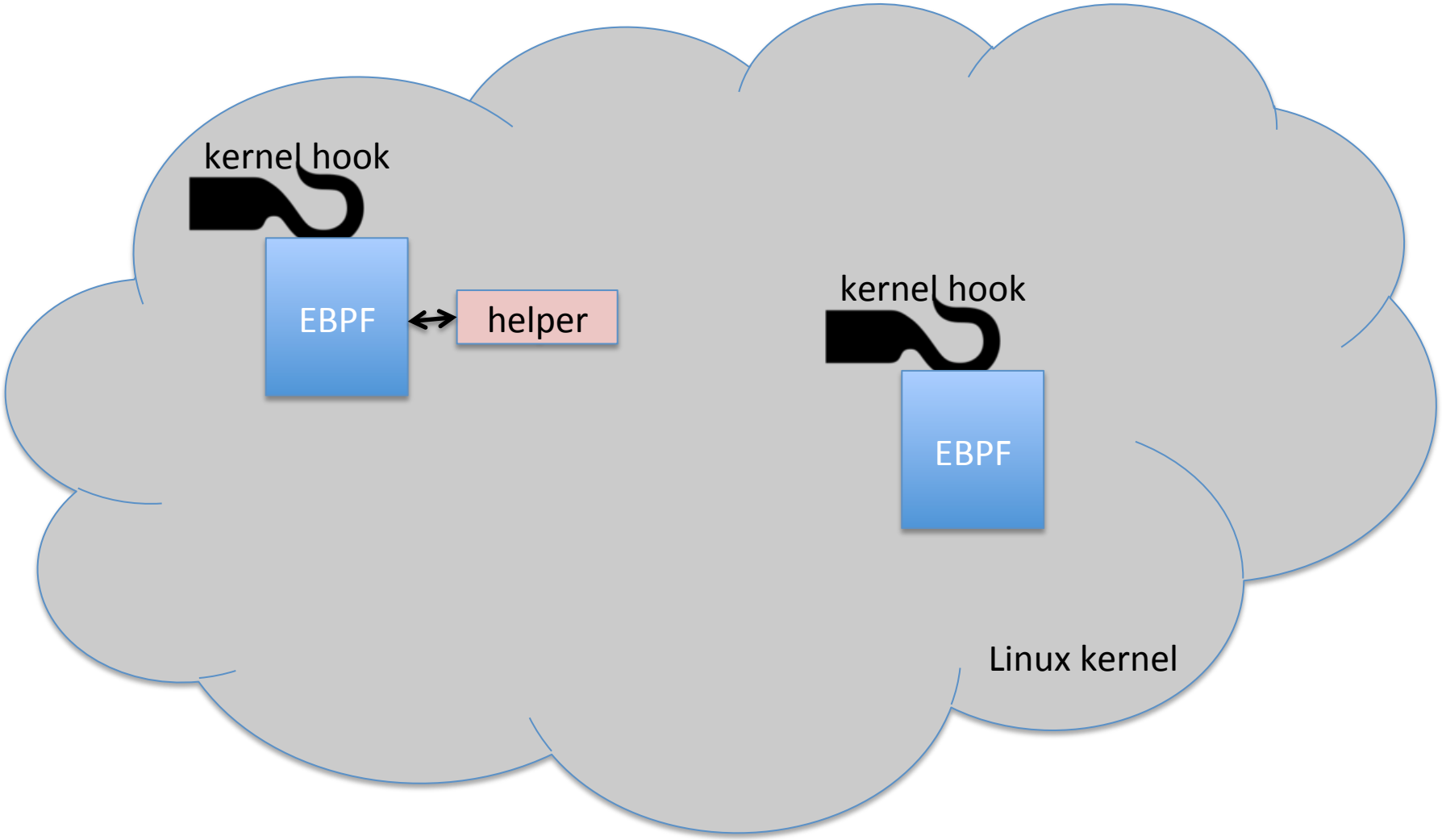
```
// switch.hpp: written by manufacturer
struct MetaIn { int inputPort; }
struct MetaOut {
    int outputPort;
    bool drop;
}


template<class T> class switch
{
    virtual void parser(const packet &p, T& headers)=0;
    virtual void pipe(T& headers,
                      const MetaIn &in, MetaOut& out)=0;
}
```



Architecture-specific black-boxes

```
// target.hpp: written by manufacturer
class Ckecksum
{
   Checksum();
   void clear();
   template<class T> void append(const T& data);
   u16 getChecksum();
}
```

EBPF's world



Parallels

- P4 switch architecture = EBPF set of hooks
- P4 black boxes = EBPF helper functions
- P4 intrinsic metadata = EBPF hook “signatures”

A suggestion

- Structured interfaces in IOVisor:
  - Header file:
    - List of predefined data types (e.g. skb, TC_*)
    - Prototypes for all EBPF hooks
    - A way to name hooks in a uniform way (TC, socket, network interface, etc.)
    - An abstracted way to attach EBPF code to a hook
    - A list of helper declarations
  - Text file: model description explaining how data flows between hooks