



CS344

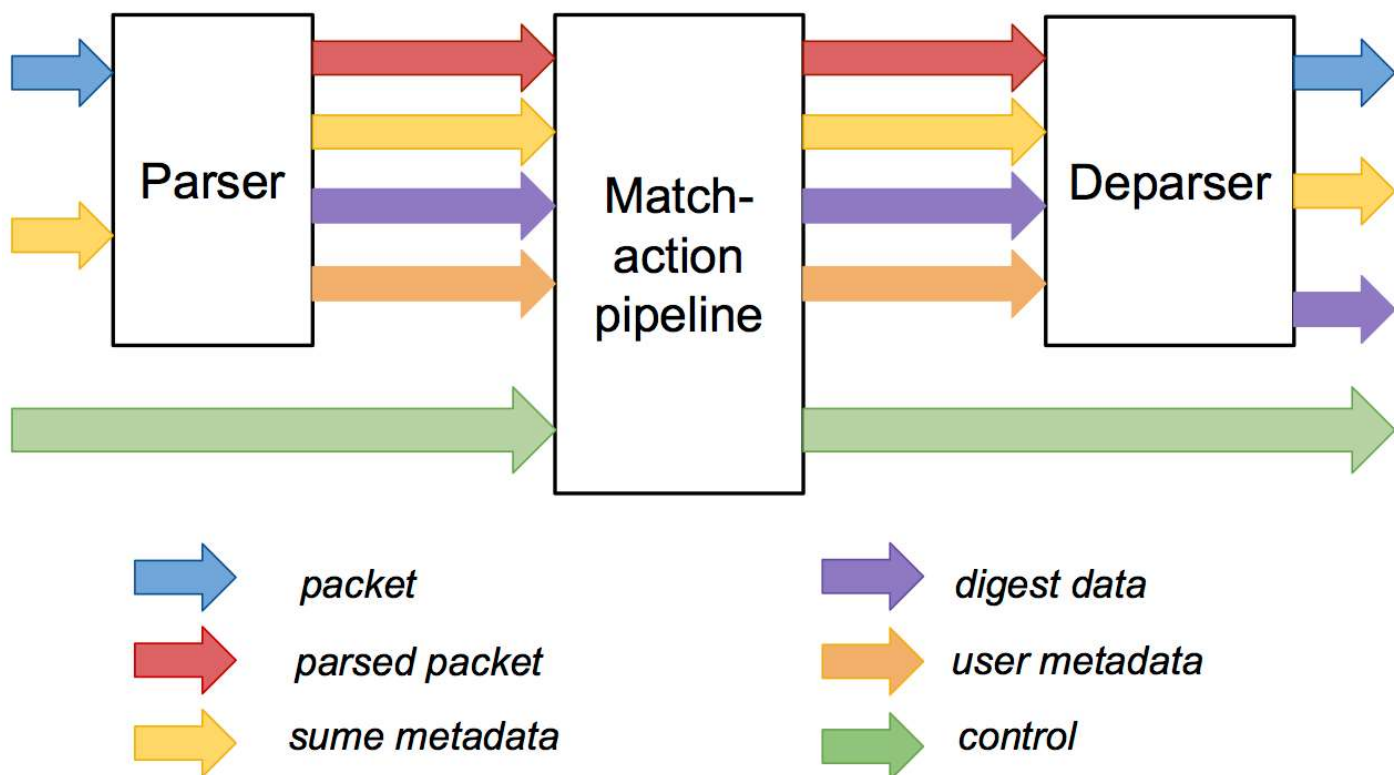
Construir um roteador de Internet

Sobre Documentação Política Cronograma Código Fonte
Equipes Palestras Piazza

Arquitetura SimpleSumeSwitch

Utilizaremos a versão mais recente da arquitetura SimpleSumeSwitch, que ainda não foi totalmente incorporada ao repositório upstream P4-NetFPGA-live. Portanto, a documentação da arquitetura SimpleSumeSwitch no wiki público P4-NetFPGA está um pouco desatualizada. Consulte este documento para obter detalhes sobre o funcionamento do SimpleSumeSwitch.

O SimpleSumeSwitch é a arquitetura P4 atualmente definida para o NetFPGA SUME. A descrição da arquitetura pode ser encontrada em `/opt/Xilinx/SDNet//data/p4include/sume_switch.p4` ou onde quer que você tenha instalado o Xilinx SDNet. A arquitetura consiste em um único analisador, um único pipeline de ação de correspondência e um único desanalisador. Conforme mostrado abaixo:



- **sume_metadata:** corresponde ao barramento tuser no projeto SUME `reference_switch`, é definido da seguinte forma:

```
struct sume_metadata_t {
    bit<16> dma_q_size; // measured in 32-byte words
    bit<16> nf3_q_size; // measured in 32-byte words
    bit<16> nf2_q_size; // measured in 32-byte words
    bit<16> nf1_q_size; // measured in 32-byte words
    bit<16> nf0_q_size; // measured in 32-byte words
    bit<8> send_dig_to_cpu; // send digest_data to CPU
    bit<8> drop; // * DEPRECATED *
    port_t dst_port; // one-hot encoded (see below)
    port_t src_port; // one-hot encoded (see below)
    bit<16> pkt_len; // (bytes) unsigned int
}
```

The format of the `dst_port` and `src_port` fields is as follows:

bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1
(nf3_dma) - (nf3_phy) - (nf2_dma) - (nf2_phy) - (nf1_dma) - (nf1_phy) - (nf0_dma) - (

- `pkt_len`- o tamanho do pacote (não incluindo o preâmbulo Ethernet ou FCS) em bytes.
- `src_port`- a porta na qual o pacote chegou. Por exemplo, se o pacote chegou na porta `nf1`, este campo seria definido como `0b00000100`.
- `dst_port`- deve ser definido pelo programa P4 do usuário para indicar de qual porta ou portas (se houver) o pacote deve ser enviado. Por exemplo, para enviar uma cópia do pacote pelas portas `nf0` e `nf2`, este campo deve ser definido como `0b00010001`.
- `drop`- este campo está obsoleto e será removido em uma versão futura. Para descartar um pacote, defina `dst_port = 0`.
- `send_dig_to_cpu`- O `digest_data` será sempre anexado ao pacote sempre que ele for encaminhado via DMA para a CPU. No entanto, se o programador P4 quiser enviar apenas o *para* `digest_data` a CPU e não o pacote inteiro, ele deve definir o bit menos significativo deste campo e garantir que todos os bits da porta da CPU do `dst_port` campo estejam definidos como 0.

- **_q_size**- o tamanho de cada fila de saída, medido em termos de palavras de 32 bytes (arredondadas para cima). Este é o tamanho das filas de saída quando o pacote *começa* a ser processado pelo programa P4.
- **digest_data**: o formato deste barramento é definido pelo programador P4. A única restrição é que ele *deve* ter 256 bits de largura. Esses dados serão anexados ao pacote quando ele for enviado via DMA para a CPU.
- **user_metadata**: o formato deste barramento também é definido pelo programador P4. Ele pode ser usado para passar informações adicionais entre o analisador, o pipeline de M/A e o desanalisador.
- **controle de entrada/saída**: esses sinais são usados para adicionar/remover entradas de tabelas e registros de controle de leitura/gravação.

Uma nota sobre nomes de interface

Muitos novos usuários costumam se confundir com os nomes das interfaces. A placa NetFPGA SUME possui 4 portas SFP+ (também conhecidas como interfaces físicas). Costumamos chamar essas interfaces de `nf0`, `nf1`, `nf2`, e `nf3`. Onde `nf0` fica a porta mais próxima das luzes de link na placa SUME? Há um bit nos campos "`src_port`" e "`dst_port`" para cada uma dessas portas (bits 0, 2, 4 e 6).

Se você digitar `ifconfig` na máquina host Linux após programar o FPGA, deverá obter algo como a saída mostrada abaixo. Os `nf0`, `nf1`, `nf2` e `nf3` mostrados aqui significam algo muito diferente do que foi explicado acima. Essas interfaces de rede são os meios pelos quais a máquina host pode se comunicar com o plano de dados no FPGA. Há também um bit para cada uma dessas interfaces nos campos `src_port` e `dst_port` (bits 1, 3, 5 e 7). Assim, por exemplo, se o plano de dados deseja enviar um pacote para o host e recebê-lo na `nf0` interface de rede Linux, ele deve definir o bit 1 do `dst_port` campo (por exemplo `dst_port = 0b00000010`,).

```
$ ifconfig
.
.
.
nf0      Link encap:Ethernet  HWaddr 02:53:55:4d:45:00
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

RX bytes:0 (0.0 B) TX bytes:300 (300.0 B)

nf1

Link encap:Ethernet HWaddr 02:53:55:4d:45:01
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:300 (300.0 B)

nf2

Link encap:Ethernet HWaddr 02:53:55:4d:45:02
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:300 (300.0 B)

nf3

Link encap:Ethernet HWaddr 02:53:55:4d:45:03
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:300 (300.0 B)