

Архитектура eBPF

Приведенный ниже включаемый файл (include) с переведенными на русский язык комментариями содержит определения пакета eBPF в одноименной модели. Файл eBPF.p4 размещается в каталоге r4include пакета r4c и доступен по [ссылке](#).

```
/*
Copyright 2013-present Barefoot Networks, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*/

#ifndef _EBPF_MODEL_P4_
#define _EBPF_MODEL_P4_

#include <core.p4>

/**
 * Массив счетчиков представляет собой плотный (dense) или разреженный
 * (sparse) массив 32-битовых целых чисел без знака, видимый плоскости
 * управления как отображение EBPF (массив или хэш). Каждый счетчик
 * адресуется 32-битовым индексом. Счетчики могут инкрементироваться
 * лишь плоскостью данных, но могут считываться и сбрасываться
 * плоскостью управления.
 */
extern CounterArray {
    /** Выделяется массив счетчиков.
     * @param max_index Максимальный поддерживаемый индекс счетчика.
     * @param sparse Массив счетчиков считается разреженным. */
    CounterArray(bit<32> max_index, bool sparse);
    /** Инкрементирование счетчика с заданным индексом. */
    void increment(in bit<32> index);
    /** Увеличение значения счетчика с заданным индексом. */
    void add(in bit<32> index, in bit<32> value);
}

/**
 * Каждая таблица должна иметь свойство реализации array_table или hash_table.
 */

/**
 * Свойство реализации таблиц, показывающее, что таблицу нужно реализовать с
 * использованием отображения в массив EBPF. Однако для таблиц с сопоставлением
 * LPM свойство реализации служит лишь для размера и используемая таблица
 * является префиксным деревом LPM (trie).
 */
extern array_table {
    /// @param size Максимальное число записей в таблице.
    array_table(bit<32> size);
}

/**
 * Свойство реализации для таблиц, показывающее, что таблицу нужно реализовать с
 * использованием отображения в хэш EBPF.
 */
extern hash_table {
    /// @param size: maximum number of entries in table
    hash_table(bit<32> size);
}

/* Архитектурная модель для архитектуры фильтрации пакетов EBPF */

parser parse<N>(packet_in packet, out N headers);
control filter<N>(inout N headers, out bool accept);

package eBPFFilter<N>(parse<N> prs,
                    filter<N> filt);

#endif
```