

1、通用软件赛题背景

网络芯片是交换机、路由器等通信设备中的核心组件，主要包括协议头解析、报文修改，以及协议头组包三大功能模块。协议头解析通过解析报文头中的字段识别协议头类型，并将协议头中的数据搬运到芯片内存当中；报文修改模块从芯片内存中取出数据进行修改，并将修改后的数据回写到芯片内存中；协议头组包模块从芯片内存中取出数据重新封装为新的报文后输出。本课题研究网络芯片中的协议头组包问题。

3、赛题简介

网络通信中数据包的包头由很多协议头数据拼接而成，每一种协议头后都只能紧跟某几种固定的协议头类型。因此，协议头的解析过程可以通过一个有限状态机描述，例如图1的有限状态机描述了以太网帧解析器。协议头组包需要满足解析状态图的拓扑顺序，即状态图中从起始节点到终止节点的任意一条路径在组包后的相对顺序不能发生改变。

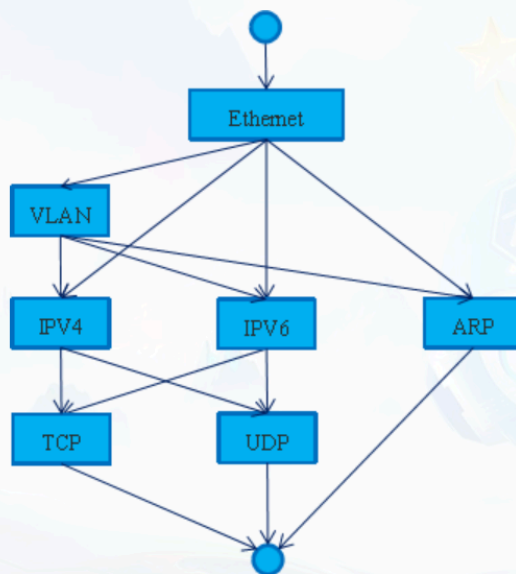


图1 以太网帧解析器有限状态机

为完整阐述协议头组包问题，首先对网络芯片的内存结构和组包机制进行介绍。

3.1芯片内存结构

芯片中的内存总共4K bit，由220个容器构成，包括64个1字节容器、88个2字节容器、68个4字节容器。将220个容器从0到219进行编号，其中0~63为1字节容器，64~151为2字节容器，152~219为4字节容器。

4K bit总共为512B，内存按字节编号为0~511。

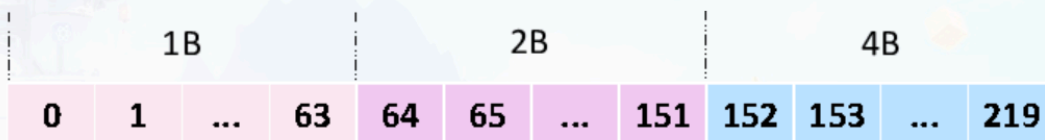


图2 芯片内存结构

3.2芯片组包机制

芯片在组包时通过域字典实现，域字典可以看作深度一定的表项，表项中的每一行为一个条目，其结构如图3所示。

2B	2B
2B	2B
...	...

域字典每个条目总共4B，由两个2B构成。将芯片的4K内存按照2B为粒度进行切分，如图4所示，域字典条目中的每个2B可对内存切分后的任意一个2B。

芯片在进行协议头组包时，会依次扫描域字典中的每个条目，条目无效直接跳过，否则将条目中两个2B取出来进行组装，需要注意的是，每个2B中有2B均有效、其中1B有效、2B均无效三种情况。2B有效时2B数据均参与组装，其中1B有效时仅有效的1B参与组装，2B均无效时2B均不参与组装。按照上述方式扫描域字典每一个条目后，即完成协议头组包功能。

2B			2B		2B		2B	2B	2B		2B	2B
0	1	...	62	63	64	...	151	152	...		219	

4、题目具体要达到的结果和规则

协议头组包问题可进一步分为两个子问题，第1个问题为协议头内存分配问题，第2个问题为域字典条目生成问题。

4.1协议头内存分配问题

协议头内存分配问题需要将状态图中的每个协议头分配到芯片内存中。尽管协议头的宽度可能会大于32bit，但本问题的输入中已将每个协议头拆分为位宽不超过32 bit的字段且拆分后的每个字段一定是整字节（即8 bit\16 bit\24 bit\32 bit），输入状态图中的每个节点对应一个字段。协议头分配问题即将每个字段分配到芯片内存空间。

需要满足的约束如下：

- 1)状态图中任意一条路径上的所有字段，其中任意两个字段在内存中分配的位置不能冲突（即分配的bit上不发生重叠）；
- 2)每个字段分配的内存不能跨容器，即每个字段只能分配到一个容器中；
- 3)字段可以分配到比其位宽大的内存容器，如8bit字段分配至2B容器中，此时只占其中1B内存空间，4B容器同理；
- 4)每个字段分配的内存需要字节对齐，即分配到1B容器时起始位置只能为0 bit，分配到2B容器时起始bit只能为0 bit或8 bit，分配4B容器时起始bit只能为0 bit、8 bit、16 bit、24 bit，因此每个字段分配后均占用整数个字节的内存空间。

4.2域字典生成问题

域字典生成是基于协议头内存分配的基础上进行。域字典生成时需要满足状态图中每条路径上解析的字段在组包时的顺序保持一致，该约束我们称为拓扑约束。不考虑多个字段复用域字典条目时，域字典生成比较简单，通过对状态图的拓扑排序即可实现，即首先将状态图进行拓扑排序，得到字段的排列顺序，然后按照排序字段分配的内存地址依次生成条目即可。该方法能够保证生成的域字典一定满足拓扑约束，但域字典资源利用率较低。事实上如果两个字段不在状态图的一条路径上，两个字段可能分配相同的内存地址，此时两个字段在域字典中可以复用条目。如图5(a)所示的简单状态图，因为字段B和字段C不在一条路径上，因此字段B和C可以分配到相同的内存地址。假设A、B、C的位宽均为4B，A分配的内存字节为0、1、2、3，B和C分配的内存字节均为4、5、6、7。不共享条目时生成的域字典如图5(b)所示，占用3个条目，共享条目时生成的域字典如图5(c)所示，仅占用2个条目。在芯片实际执行时，如果解析路径为A->B，则组包时A和B有效，在图5(b)中的域字典中，组包时将第1条和第2条取出组包；如果解析路径为A->C，则组包时A和C有效，取出第1条和第3条组包。在图5(c)的域字典中，无论解析哪条路径，均取出第1条和第2条组包，同样可以实现正确组包。

域字典生成需尽量最大化共享条目，以实现芯片资源最大化利用。

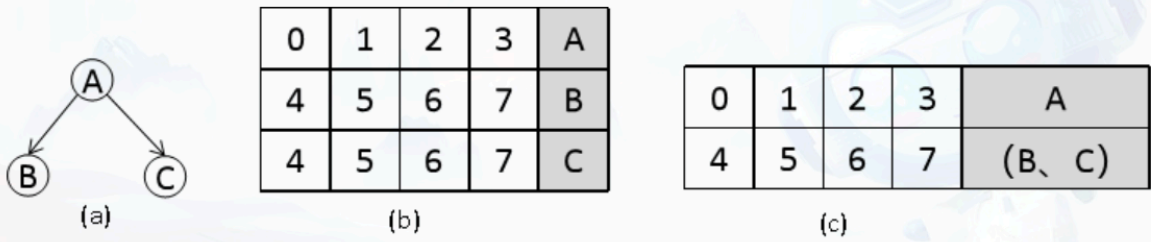


图5 状态图示例

域字典条目生成问题的约束包括：

拓扑约束：状态图中每条路径上解析的字段在组包时的顺序均保持一致；

每个字段对应的内存只能占用一个条目；

不能存在两个条目完全相同；

每个条目有4B，可以1B、2B、3B、4B有效，无效的字节使用占位符“-”表示。由于4B由内存中的2个2B构成，因此具体约束如下：

- (1)第1B或第3B有效时必须是偶数字节；
- (2)第2B或第4B有效时必须是奇数字节；
- (3)如果第1B和第2B均有效，则第1B的字节与第2B的字节连续；
- (4)如果第3B和第4B均有效，则第3B的字节与第4B的字节连续。
- (5)第2B和第3B不要求连续

该约束需要尤其注意，表1给出了部分合法与不合法的条目示例。

表1 域字典条目说明

0	1	2	3	合法, 4B 有效	
0	1	2	4	不合法, 第 3 字节和第 4 字节不连续	
0	1	3	2	不合法, 第 3 字节为奇数	
0	1	2	-	合法, 3B 有效	
0	1	-	2	不合法, 第 4 字节为偶数	
0	1	-	3	合法, 3B 有效	
0	-	2	3	合法, 3B 有效	
1	-	2	3	不合法, 第 1 字节为奇数	
0	-	3	5	不合法, 第 3 字节和第 4 字节不连续	
0	1	-	-	合法, 2B 有效	
-	7	8	-	合法, 2B 有效	
7	8	-	-	不合法, 第 1 字节为奇数	
0	-	-	-	合法, 1B 有效	
-	1	-	-	合法, 1B 有效	
-	0	-	-	不合法, 第 2 字节为偶数	

5、优化目标

针对协议头内存分配问题，优化目标为最小化占用的内存字节数；针对域字典生成问题，优化目标为最小化域字典条目的数量。

7、赛题数据

1) 输入

输入文件input.csv给出了每个字段的位宽，以及按照邻接表的形式给出了状态图，其中每行第1个元素为字段编号，第2个元素为该字段的位宽，单位为比特(bit)，后续元素为该字段在状态图中的邻接字段。格式：

字段编号, 位宽, 邻接节点编号 (0个、一个或者多个)

2) 输出

输出包括两个文件，分别命名为output1和output2.csv。

output1.csv为各字典分配内存位置。其中每一行第1个元素为字段编号，后续元素为该字段分配的内存地址的字节编号（字节编号按从小到大排列，由于每个字段最大32 bit，因此最大分配4字节，分配字节不能跨容器），格式：

字段编号1, 字节号1, 字节号2, 字节号3, 字节号4 (1个-4个)

字段编号2, 字节号1, 字节号2,

... ..

ouput2.csv为域字典。其中每一行为域字典的一个条目，无效字节用占位符“-”表示，格式如表1中蓝色部分所示，注意需要保证条目有效性，格式：

字节编号1,字节编号2,字节编号3,字节编号4, 该条目对应的字段编号。

以表1 结果, 格式举例:

0,1,2,3,字段编号 (可多个,多个时用“,”隔开)

0,1,2,4,字段编号 (可多个,多个时用“,”隔开)

0,2,3,4,字段编号 (可多个,多个时用“,”隔开)

0,1,3,2,字段编号 (可多个,多个时用“,”隔开)

0,1,2,-,字段编号 (可多个,多个时用“,”隔开)

0,1,-,3,字段编号 (可多个,多个时用“,”隔开)

0,-,23,字段编号 (可多个,多个时用“,”隔开)

0,1,-,-,字段编号 (可多个,多个时用“,”隔开)

-,7,8,-,字段编号 (可多个,多个时用“,”隔开)

7,8,-,-,字段编号 (可多个,多个时用“,”隔开)

0,-,-,-,字段编号 (可多个,多个时用“,”隔开)

-,1,-,-,字段编号 (可多个,多个时用“,”隔开)

-,0,-,-,字段编号 (可多个,多个时用“,”隔开)

