

# 基于 ABNF 的高速硬件解码方法的研究

屈玉贵, 李奇越, 赵保华, 刘桂英

(中国科学技术大学电子工程与信息科学系, 安徽合肥 230027)

**摘要:** 本文提出了一种硬件实现网络协议解码的方法. 这种方法采用基于流水线的高速并行字符串匹配结构和基于 ABNF(Augmented Backus Naur Form)范式定义的高速域值合法性检查. 文中还对用这种方法实现的 SP(Session Initiation Protocol)协议解码器的性能进行了分析, 并与软件实现进行了对比. 通过比较可以得出, 采用本方法所实现的协议解码系统只需要较少的硬件资源就可以得到比传统的软件实现几十倍的性能提高, 并显著提高整个系统的吞吐量.

**关键词:** 流水线; ABNF 范式; 字符串匹配; 合法性检查

**中图分类号:** TN47 **文献标识码:** A **文章编号:** 0372-2112 (2006) 09-1645-04

## The Research on High-Speed Hardware Decoding Based on ABNF

QU Yu-gui, LI Qi-yue, ZHAO Bao-hua, LU Gui-ying

(Department of Electronic Engineering & Information Science, USTC, Hefei Anhui 230027, China)

**Abstract** A method to implement high-throughput network protocol decoder using hardware is proposed. Two technologies are used in this method: high speed parallel string match based on pipeline and high speed message validity check based on ABNF(Augmented Backus Naur Form). The performance of SP(Session Initiation Protocol) protocol decoder implemented by this method is also evaluated and the comparison with software implementation is made. It can be found out that the network protocol decoder implemented by this method can get an improvement up to tens of times while only relatively little hardware resource is needed and the throughput of the system can be improved dramatically.

**Key words** pipeline; Augmented Backus Naur Form (ABNF); string match; pattern match

### 1 引言

采用 ABNF 范式可以方便完整的定义一套网络协议, 因此在 Internet 定义中 ABNF 被广泛使用, 但在对网络协议解码的过程中, 传统的软件实现解码速度缓慢, 尤其是对于需要大量文本字符串查找和匹配操作的解码正确性检查部分, 更是耗费了 CPU 时间, 在 SIP 服务器中, 当服务器负荷较重的时候, 仅仅是解码工作就会占用服务器 CPU 近 40% 的资源. 本文提出了一种使用硬件实现网络协议解码的方法. 在本结构中, 采用了基于流水线的字符串匹配操作来标记协议报文中关键字 (keywords) 的位置并进行域值的提取, 再根据协议的 ABNF 范式定义, 用硬件逻辑结构实现域值的正确性检查部分. 采用本文所提出的方案, 可以对任何一种采用 ABNF 范式定义的文本类型协议进行解

码, 并且速度上与软件相比有极大的提高.

由于硬件的并行性和充足的资源使得其非常适合于实现文本字符串查找和匹配操作. 文 [5, 6] 中提出了一种用硬件实现 KMP 算法的方法来进行字符串的查找, 虽然 KMP 算法在字符串查找方面具有很大的优势, 但是这种算法采用的是串行搜索的方式, 没有充分利用硬件的并行性, 所以并不适合在硬件协议解码器中使用. 在本文中采取了一种基于流水线的高速字符串匹配方法, 来进行对 keywords 的匹配, 并提取相应的域值. 文 [1, 2] 中提出了用硬件逻辑来构造正则表达式的方法, 可以将三种最基本的正则表达式用的硬件逻辑表示, 但 ABNF 中定义的范式类型有所扩展, 用文 [1] 中的三种结构不能表示所有的 ABNF 范式. 本文将文 [1] 中的三种结构扩展为四种, 用此四种基本结构就可以描述所有的 ABNF 范式, 并在此基础上

收稿日期: 2005-01-11; 修回日期: 2006-07-06

基金项目: 国家自然科学基金重大研究计划项目 (No. 90104010); 国家自然科学基金项目 (No. 60241004); 国家 973 计划项目 (No. 2003CB314801); 网络与交换技术国家重点实验室开放课题

用硬件逻辑构造出解码器中的域值的合法性检查模块。

### 2 高速网络协议报文解码系统结构

在每一个 ABNF 范式定义的文本文字类型的网络协议报文中都含有一些固定的字符和字符串用来标志报文中特定的域的存在或者标志报文的类型。例如在 SIP 协议中, 有如下 ABNF 定义:

```
SIP-URL = " SIP: " [ userinfo "@" ] hostport url-parameters
[ headers ]
userinfo = [ username ] [ ":" password ]
url-parameters = * ( ";" param-name "=" param-value )
```

从上述 ABNF 定义可以看出, 在一条定义 SIP-URL 的规范中, 有 "SIP:" "@": ":" ";" "=" "&" 等字符和字符串来标志 username password hostname hostport param header 等域是否存在。因此, 在域值提取的时候就可以通过查找并记录这些关键字和字符串 (keyword) 在报文中的位置, 就可以根据这些 keyword 的位置计算出需要提取域值的域是否存在以及这些域在报文中的位置。

ABNF 范式中不仅定义了报文的格式, 还定义了报文中每个域所包含的合法的字符。例如:

```
username = * ( unreserved | escaped | "&" | "=" | "+" | "$" | ";" | ":" | "?" | "/" )
escaped = "%" hex hex
mark = "-" | "_" | "." | "!" | "~" | "*" | "(" | ")"
```

上述几条范式定义规定了 username 域所能包含的字符, 其他的字符如 "\*" 等都属于非法字符。如果在报文中含有这样的非法字符, 系统必须能够给出出错信息, 防止整个报文提交到协议上层作进一步的处理。本文利用硬件并行处理的特点, 构造了一个能够依据 ABNF 范式对提取出的域值进行合法性检查的系统。

系统结构如图 1 所示:

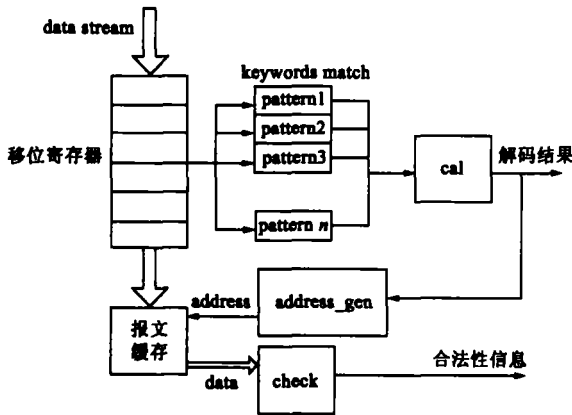


图 1 高速网络协议解码器系统结构

在图 1 中, 输入数据首先进入一组移位寄存器再缓存到一个存储器中, 并同时与多个 keyword 进行并行比较, cal 模块根据比较得出的 match 信号就可以计算出在协议报文中

中存在的域以及这些域值在报文中的位置。cal 模块在输出解码结果的同时, 还把这些结果送到 address\_gen 模块中, 此模块的作用是根据 cal 模块计算出的域值的位置产生报文缓存的地址, 用以把相应的域值再次提取出来便于检查报文的合法性。提取出的域值输入到 check 模块中, 此模块根据协议的 ABNF 范式构造了合法性检查的硬件逻辑, 通过 check 模块对域值进行检查之后, 就可以输出整个报文的合法性信息。

### 3 硬件设计中的关键技术

为了提高吞吐率和系统时钟频率, 将整个系统设计按照功能划分为 4 个模块, 流水线组成如图 2 所示。

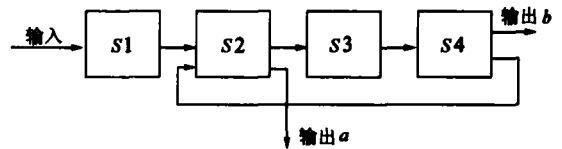


图 2 4 段流水线组成方式

S1 字符串匹配模块, 对当前输入的报文字符与 keyword 进行比较产生匹配信号。

S2 位置计算模块, 根据 S1 输出的匹配信号计算需要提取的域值在原报文中的开始和结束位置。

S3 地址生成模块, 根据 S2 的输出生成需要检查的域值在报文缓存中的存放地址。

S4 合法性检查模块, 以 S3 的输出作为地址, 从报文缓存中提取报文数据, 根据相应域的 ABNF 范式定义进行合法性检查, 并输出合法性信息。

输出 A 是对输入报文解码之后提取出的域值在原报文中的开始和结束位置。输出 B 是对每一个域进行合法性检查之后得到的报文合法性信息。

整个流水线的输入是以行为单位的报文数据, 假设一行平均有 k 个字符, m 个域需要提取, 每个域有 n 个字符, 显然有  $1 \leq m * n \leq k$ , S1 完成对这一行报文的处理需要 k 个时钟周期, S2 需要 1 个时钟周期, S3, S4 各需要 n 个时钟周期, 从 S4 到 S2 有 m 次反馈。那么此流水线的预约表如表 1 所示。

表 1 4 段流水线预约表

	$t_0$	$t_1$	$t_2$	$t_3$	...	$t_{m+1}$	$t_{m+2}$
S1	AB				...		AB
S2		AB			...		
S3			B		...		
S4				B	...	B	

如果不采用流水线操作, 处理 p 行报文需要的总时间为

$$T_{unpipe} = [k + (1 + n + n) * m] * p$$

采用流水线操作后, 处理 p 行报文需要的总时间为

$T_{pipe} = [k + (1 + n + n) * m] * p$ , 因此, 当系统时钟频率不变时, 采用流水线操作带来的加速比为

$$S = \frac{T_{\text{unpipe}}}{T_{\text{pipe}}} = \frac{[k + (1 + n + n) * m] * p}{k + (1 + n + n) * m + k(p - 1)}$$

当  $p$  很大时,  $S \rightarrow \frac{k + 2nn + m}{k}$ , 对于类似于 SIP 等大多数应用层协议, 需要提取的域值内容较多,  $m * n \approx k$ , 所以, 加速比  $S \approx 3$

#### 4 基于流水线的并行字符串匹配

由系统功能分析可以看出, 在域值提取部分中最主要的部分就是字符串匹配. 在本系统中使用了一组移位寄存器来实现高效率的字符串匹配, 以匹配字符串“SIP:”为例, 结构如图 3 所示:

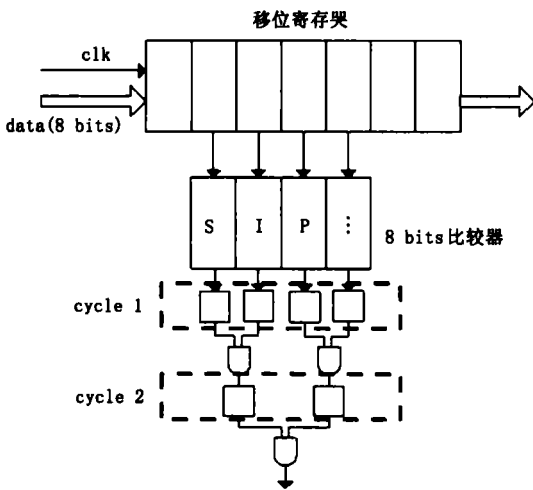


图 3 基于流水线的并行字符串匹配

在上述结构中, 采用  $N$  个比较器, 每个时钟周期就可以对  $N$  个字节的数据进行处理, 而且对所有需要匹配的字符和字符串的比较都是并行执行的. 虽然这种结构所占用的硬件资源是随着被匹配的字符串的增多而线性增加的, 但是整个系统的性能并不会随之降低——在长度为  $n$  的字符流中查找长度为  $m$  的字符串只需要  $n$  个时钟周期. 因此, 在对网络协议解码系统中, 采用这样的字符串匹配方法是非常合适的.

#### 5 基于 ABNF 的域值合法性检查

根据 ABNF 语法规则, 可将基本规则总结为如下四种:

- 连接规则:  $rule = subrule1\ subrule2$
- 选择规则:  $rule = subrule1\ | \ subrule2\ | \ subrule3$
- 循环规则:  $rule = * \ (subrule)$
- 可选规则:  $rule = subrule1\ [ \ subrule2 \ ] \ subrule3$

其他各种复杂的范式定义都可以由这四种基本规则组合而成, 所以在硬件实现的时候也实现了这四种基本的结构, 然后在其之上组合构建其他各种结构.

文献 [1] 中提出了用硬件逻辑来构造三种最基本的 NFA 表达式, 但 ABNF 中定义的范式类型有所扩展, 用文

[1] 中的三种结构不能表示所有的 ABNF 范式. 本文将文 [1] 中的三种结构扩展为四种, 如图 4 所示:

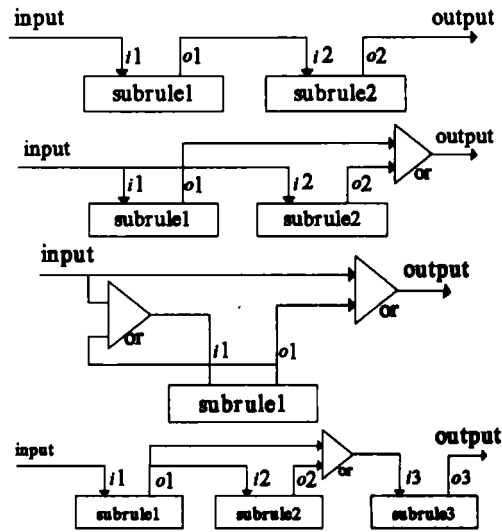


图 4 四种 ABNF 基本关系式的硬件实现

基于以上四种基本结构, 就可以实现各种复杂的 ABNF 范式定义. 例如, 规则  $username = * \ (unreserved\ | \ escaped\ | \ other\_char)$  用上述硬件逻辑实现如图 5 所示:

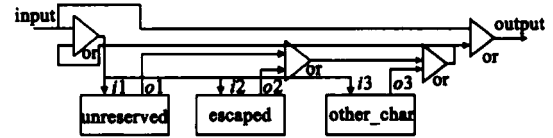


图 5 规则 username 的硬件逻辑实现结构

#### 6 性能与实现成本分析

为了对系统性能进行测试, 本节用 VHDL 语言描述实现了 SIP 协议解码的硬件系统, 并针对 Altera 公司的 Cyclone EP1C12Q240-8 器件, 用 Quartus 4.1 进行综合布线, 同时还用软件实现了 SIP 协议报文解码, 并在普通 PC 机、实际的路由器中进行测试并记录软件处理所需要的时间.

各种实现方案的性能比较如图 6 所示. 其中 PC 机的硬件配置为: Intel P4 2.4G CPU, 512MB 内存; 路由器的硬件配置为 PowerPC750 512MB 内存.

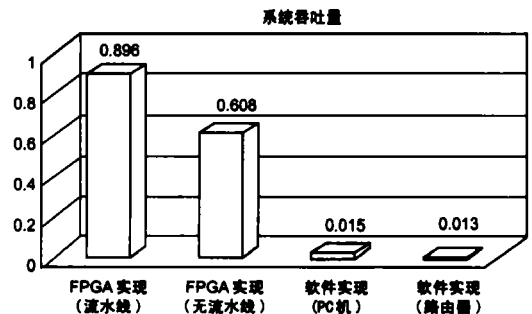


图 6 SIP 协议解码各种实现的性能比较

为了测试硬件协议解码器带来的成本提高,本节还对在 NGN (Next Generation Network) 中广泛使用的协议进行了测试. 测试标准是各个协议的硬件实现的资源占用情况,如表 2 所示:

表 2 各个协议硬件解码系统资源占用

协议名称	综合频率 (MHz)	系统吞吐量 (Gbps)	逻辑单元	存储器单元 (bits)
SIP	112	0.896	5345	30208
RTSP	98	0.784	6108	34523
RADIUS	116	0.928	5160	29166
DIAMETER	124	0.992	4827	27284

从表 2 可以看出,采用本文提出的方法实现这些复杂的应用层协议的硬件解码系统只需要较少的硬件资源便可达到很高的性能. 采用一片 Cyclone EP1C12Q240-8 器件便可轻松实现以上各个协议的硬件解码系统,以一百元左右的成本提高换取解码系统几十倍的性能提升是一件相当可观的事情.

## 7 结论

本文提出了一种硬件实现高速网络协议解码的方法. 这种方法主要采用了两项技术:

- (1) 基于流水线的高速并行字符串匹配结构;
- (2) 基于 ABNF 范式定义的高速域值合法性检查.

为了提高硬件协议解码器的吞吐率,本文还将硬件设计按照功能划分为 4 段流水线,并计算了采用流水线方式之后带来的加速比;最后本文对用这种方法实现的 SIP 协议解码的性能进行了分析,并与软件实现进行了对比,通过比较可以得出,采用本方法所实现的协议解码系统只需要较少的硬件资源就可以得到比传统的软件实现几十倍的性能提高,而且可以采用能重新配置的 FPGA,可以方便的更换不同的协议,这样就可以极大的减轻路由器 CPU 的负担.

## 参考文献:

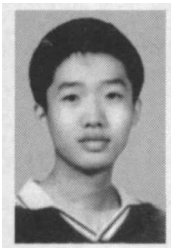
- [1] Reetinder Sidhu, Viktor K Prasanna. Fast regular expression matching using FPGAs [A]. IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2001) [C]. Los Alamitos: IEEE CS Press, April 2001: 227-238.
- [2] J Moscola, J Lockwood, R P Loui. Implementation of a content-scanning module for an internet firewall [A]. IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2003) [C]. Napa CA, USA: IEEE CS Press, Apr 2003: 31-39.

- [3] James Moscola, Michael Pachos, John Lockwood, Ronald P Loui. Fpsed: A streaming content search-and-replace module for an Internet firewall [A]. 11<sup>th</sup> Symposium on High Performance Interconnects [C]. Stanford CA, USA: IEEE CS Press, August 2003: 122-129.
- [4] Reetinder Sidhu, Viktor K Prasanna. String matching on multi-context FPGAs using self-reconfiguration [A]. ACM / SIGDA International Symposium on Field Programmable Gate Arrays [C]. New York, NY, USA: ACM Press, Feb 1999: 217-226.
- [5] Zachary K Baker, Viktor K Prasanna. Time and area efficient pattern matching on FPGAs [A]. Proceedings of the Twelfth Annual ACM International Symposium on Field Programmable Gate Arrays [C]. New York, NY, USA: ACM Press, Feb 2004: 223-232.
- [6] M Handley, et al. SIP: Session Initiation Protocol [S]. RFC2543. IETF, March 1999.
- [7] J Rosenberg, H Schulzrinne, et al. SIP: Session Initiation Protocol [S]. RFC3261. IETF, June 2002.

## 作者简介:



屈玉贵 教授, 博士生导师, 研究方向: 无线传感器网络, 协议理论和工程.  
Email: ygqu@ustc.edu.cn



李奇越 博士研究生, 研究方向: 网络体系结构, 协议理论和工程.



赵保华 教授, 博士生导师, 研究方向: 无线传感器网络, 协议理论与工程.  
Email: bhzhao@ustc.edu.cn