基于应用区域边界体系结构的安全模型*)

刘益和[1,2]　沈昌祥[3]

(内江师范学院计算机与信息科学系　内江 641112)[1]　(四川大学信息安全研究所　成都 610064)[2]

(海军计算技术研究所　北京 100841)[3]

摘　要　基于信息系统的整体安全解决，国内外信息安全专家提出了信息安全保障、信息安全体系结构等概念，其中最著名的是美国国家安全局推出的《信息保障技术框架(IATF)》3.1版和国内专家提出的"三横三纵两个中心"信息安全体系结构。前不久，陈兴蜀在这两个信息安全体系结构思想指导下，提出了应用区域边界的安全体系结构。本文在此给出了该体系结构的安全模型。通过分析，我们认为：该模型的规则是合理的、安全的。通过构建信息安全体系结构模型，将推动信息安全的理论研究。

关键词　信息安全体系结构，BLP模型，Biba模型，RBAC模型，粒度控制

A Security Model Based on Architecture Research in Application Area Boundary

LIU Yi-He[1,2]　SHEN Chang-Xiang[3]

(Department of Computer and Information Science, Neijiang Teachers College, Neijiang 641112)[1]

(Institute of Information Security, Sichuan University, Chengdu 610064)[2]

(Computing Technology Research Institute of Navy, Beijing 100841)[3]

Abstract　In order to provide the whole security solution of the information system, the domestic and foreign information security experts have put forward new concepts, such as information assurance, information security architecture, which of most famous are "This Information Assurance Technical Framework(IATF)3.1" issued by: US National Security Agency and the technological frames of information assurance, "three horizontal three longitudinal and two centers", which the domestic expert put forward. Not long ago, Chen Xing-shu put forward security architecture research in application area boundary, on the basis of the two security architecture, a security model is given out about Chen's in this paper, and by analysis, the roles in the model are reasonable and safe. There will be a active effect for information security theory research by design the model of information security architecture.

Keywords　Information security architecture, BLP model, Biba model, RBAC model, Granular control

1　引言

随着网络的广泛应用和不断发展，信息安全问题越来越受到重视。然而，对信息安全的解决方案常常采用被动的方式，发现什么地方可能有风险，有安全问题，便采取对应的安全技术来解决。这种安全技术方案缺乏对信息系统的整体考虑，故目前国内、国外的信息安全专家提出了信息安全保障、信息安全体系结构等概念，用于提供信息系统的整体安全解决方案。

国际标准化组织提出的开放系统互连参考模型(OSI/RM)起了里程碑的作用；它的第2部分安全体系结构GB/T 9387.2-1995[1]从体系结构的观点描述了OSI参考模型的安全通信必须提供的安全服务及安全机制，确立了开放互连系统的安全体系结构框架。网际互连协议(IP)安全结构(RFC 2401)[2]，定义了IPsec系统的基本结构，该结构的目的是为IP层传输提供多种安全服务。美国国家安全局在2002年9月推出了《信息保障技术框架(IATF)》3.1版[3]，为美国政府和工业界的信息与信息基础设施提供了技术指南。文[4]给出了"三横三纵两个中心"的信息安全保障技术框架，其中明确地阐述了应用环境、应用区域边界及网络传输平台等概念。前不久，陈兴蜀在IATF和"三横三纵两个中心"的体系结构思想指导下，提出了应用区域边界的安全体系结构[5]。

上述信息安全体系结构的建立，对信息安全的理论研究起到十分重要的作用。然而，对这些理论的安全性描述和证明，将是信息安全研究者面临的新课题。我们知道，信息安全是指机密性、完整性和可用性的结合，信息系统的安全策略用信息安全模型来描述，分形式化和非形式化两种。目前存在各种安全模型，如：BLP模型[6]、Biba模型[7]、RBAC(Role-Based Access Control)[8]等，它们各有特点。模型化方法为信息安全体系结构的安全性描述和证明提供了一种有效的手段。

本文的目的是：给出陈兴蜀提出的应用区域边界的安全体系结构的安全模型。

2　基本概念和模型

2.1　基本概念

计算机中存在大量涉及安全的操作。凡实施操作的称为主体，其集合用S表示；被操作的对象称为客体，其集合用O

*)国家"973"资助项目(1999035801)、四川省应用基础研究计划课题(04JY029-096)。**刘益和**　副教授、博士生，主要研究方向：信息安全；**沈昌祥**　中国工程院院士、博导，主要研究方向：信息安全体系结构、电子工程、操作系统。

表示；主客体密级(机密性等级)分别用 $T(s)$，$T(o)$表示；主客体的完整性等级分别用 $I(s)$、$I(o)$表示；并具体量化。参见表1。

表1 $T(s)/T(o)$ and $I(s)/I(O)$与 GB17859-1999 对应关系

安全级	安全描述	T(s)/T(o)	I(s)/I(o)
第1级	用户自主保护级	1	1
第2级	系统审计保护级	2	2
第3级	安全标记保护级	3	3
第4级	结构化保护级	4	4
第5级	访问验证保护级	5	5

2.2 一些已知模型的基本性质

2.2.1 BLP 模型[6]和 Biba 模型[7] 根据 BLP 模型最基本性质，系统处于安全状态需满足：如果 $T(s)\geqslant T(o)$，主体可以读客体；如果 $T(s)\leqslant T(o)$，主体可以写客体。根据 Biba 模型最基本性质：如果 $I(s)\leqslant I(o)$，则主体可以读客体；如果 $I(s)\geqslant I(o)$，则主体可以写客体。

2.2.2 RBAC 模型[8] RBAC 模型的基本思想是职责划分，这很类似一个组织机构。在 RBAC 模型中，用户被授予角色，角色被授予权限，权限关联操作。用户通过被授予的角色得到该角色的相应权限，来完成某些操作。本文以最基本 RBAC 模型的概念为准。

2.3 应用区域边界的安全体系结构[5]

应用区域边界的安全体系结构(无特殊说明，以下均简称体系结构)是针对应用环境的，它构建在端系统和区域网关的应用层(TCP/IP 协议栈的应用层)。在该体系结构中，根据处理的信息和功能划分为：应用数据层、应用协议层、安全插件层和会话连接控制层，以及贯穿于体系结构各个层次的管理控制机制。管理控制机制由两个方面构成：安全规则及访问控制，其结构如图1所示。

在应用区域边界的安全体系结构中，每层能对其上一层的信息进行封装处理。

3 基于应用区域边界的安全体系结构模型

3.1 基本概念

安全服务：针对客体 o，体系结构提供的安全服务(如加密)的集合记为：o_sa，它是体系结构的安全插件集合(记为 P)的子集。

控制粒度：当信息到达应用区域边界的安全体系结构各层时，需要进行粗粒度或细粒度控制；客体 o 的粗粒度、细粒度集合分别记为：o_rg(roughly granular control)，o_fg(finely granular control)；整个体系结构对应的粗粒度集、细粒度集合分别记为：P_rg，P_fg。

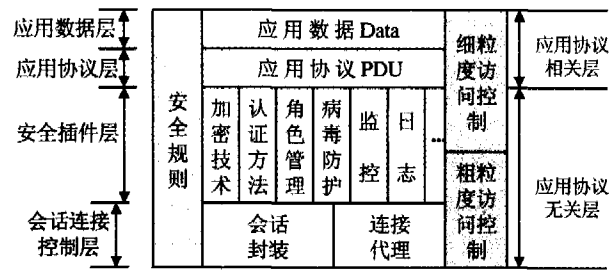

图1 应用区域边界的安全体系结构

类似地，对主体定义相应的粒度控制集合，用于描述当主体作为会话的发起者或接受者时的基本性质：客体 s 的粗粒度、细粒度集合分别记为：s_rg，s_fg，体系结构提供的安全服务(如加密)的集合记为：s_sa，它是体系结构的安全插件集合(记为 P)的子集。

不需进行粒度控制时，相应的粒度控制集合取为空集 ϕ。

下面以最基本 RBAC 模型的性质为依据，给出一些相关记号。

系统拥有的角色集合记为 R，令 $R=\{r_1,r_2,\cdots,r_n\}$；所有角色对应的权限集集合记为 R_P，令 $R\_P=\{p_1,p_2,\cdots,p_m\}$；主体 s 拥有的角色集合记为 $SR(s)$；角色 r 对应的主体集合记为 $RS(r)$；客体 o 拥有的角色集合记为 $OR(o)$；角色 r 对应的客体 o 集合记为 $RO(r)$；角色 r 对应的权限集合记为 $RP(r)$；主体 s 基于角色拥有的权限集合记为 $SP(s)$；客体 o 基于角色允许被使用的权限集合记为 $OP(o)$。

主体 s 从安全角度考虑对客体 o 的访问模式 access_model(s,o)，其值域就是 s 作为一个用户。当扮演一个角色，同时考虑 BLP 模型、Biba 模型对应的安全限制，对 o 进行操作的所有权限集合，它实际上是 R_P 的子集，其中应含 re：read，w：write，c：create；d：delete 等权限。

主体 s 从安全角度考虑对主体 s′的访问模式 access_model(s,s′)，在本文其值取 c：create；d：delete 两个。

这样，描述一个主体 s 有因素：s，T(s)，I(s)，SR(s)，s_rg，s_fg；一个客体 o 有因素：o，T(o)，I(o)，OR(o)，o_sa，o_rg，o_fg。

这些在许多文献中称为可标记主体或客体。

3.2 安全规则

3.2.1 主体创建规则

Creat_subject(s′，s：s_tlev，s_Ilev，SR(s)_lev，s_sa_lev，s_rg_lev，s_fg_lev)//主体 s′创建主体 s；s_tlev，s_Ilev，SR(s)_lev，s_sa_lev，s_rg_lev，s_fg_lev 分别表示主体 s 的密级、完整性等级、s 拥有的角色集、体系结构针对 s 的安全服务、s 的粗粒度、s 的细粒度集。

If $s'\_sa\not\subset P$ or $c\notin$ access_model(s′，s) then go_end//体系结构没有提供 s′相应的服务或 s′对 s 无创建权，结束
If $s\notin S$ then
$S=S\cup\{s\}$，T(s)＝s_tlev，I(s)＝s_Ilev，SR(s)＝SR(s)_lev，s_sa＝s_sa_lev，s_rg＝s_rg_lev，s_fg＝s_fg_lev
$\forall s''\in S$，access_model(s″，s)＝access_model(s′，s)
$\forall r\in SR(s)$，$SP(s)=\cup RP(r)$//构造主体的权限集，即为所有 RP(r)的并集
$\forall o\in O$，access_model(s，o)＝$SP(s)\cap OP(o)$
DO CASE //对任意的 o，将 T(s)，T(o)，I(s)，I(o)分情况讨论，构造 access_model(s，o)
CASE (T(s)＞T(o) and I(s)≤I(o)) or (T(s)＝T(o) and I(s)＜I(o))
If {w}∈access_model(s，o) then access_model(s，o)＝access_model(s，o)－{w} // 去掉 w 权利，转结束。
CASE (T(s)＜T(o) and I(s)≥I(o)) or (T(s)＝T(o) and I(s)＞I(o))
If {re}∈access_model(s，o) then access_model(s，o)＝access_model(s，o)－{re}// 去掉 re 权利，转结束。
CASE (T(s)＞T(o) and I(s)＞I(o)) or (T(s)＜T(o) and I(s)＜I(o)) If {re，w}∈access_model(s，o) then access_model(s，o)＝access_model(s，o)－{re，w}// 去掉 re，w 权利，转结束。
ENDCASE

3.2.2 创建客体规则

Creat_object(s′，o：o_tlev，o_Ilev，OR(o)_lev，o_sa_lev，o_rg_lev，o_fg_lev)// 主体 s′创建客体 o，这里 o_tlev，o_Ilev，OR(o)_lev，o_sa_lev，o_rg_lev，o_fg_lev 分别表示客体 o 的密级、完整性等级、客体 o 允许被使用的角色集、

体系结构针对客体 o 使用的安全服务集、客体 o 的粗粒度、客体 o 的细粒度集合。

If s′ sa⊄P or c∉SP(s′)then go end//体系结构没有提供 s′ 相应的服务或 s′对 o 无创建权，结束
If o∉() then ()=()∪{o}，T(o)=o_tlev，
I(o)=o Ilev，OR(o)=OR(o)_lev，
o sa=o sa_lev，o rg= o rg_lev，
o fg=o_fg lev
∀r∈OR(o)，OP(o)=∪RP(r)//构造客体的权限集合，所有 RP(r)的并集。
∀s∈S，access model(s，o)=SP(s)∩OP(o)//从角色的角度考虑，只有 s 拥有某项权限，而这一权限，允许对客体 o 使用时，主体 s 才能访问客体 o。后面的函数类似。
DO CASE //对任意的 s，将 T(s)，T(o)，I(s)，I(o)分情况讨论，构造 access model(s，o)
CASE (T(s)>T(o)and I(s)≤I(o))or (T(s)=T(o)and I(s)<I(o))
If {w}∈access model(s，o) then access_model(s，o)=access_model(s，o)−{w} // 去掉 w 权利，转结束。
CASE (T(s)<T(o)and I(s)≥I(o))or (T(s)=T(o)and I(s)>I(o))
If {re}∈access model(s，o) then access_model(s，o)=access model(s，o)−{re}// 去掉 re 权利，转结束。
CASE (T(s)>T(o)and I(s)>I(o))or (T(s)<T(o)and I(s)<I(o))
If {re，w}∈access_model(s，o)then access_model(s，o)=access_model(s，o)−{re，w} // 去掉 re，w 权利，转结束。
ENDCASE

3.2.3 经体系结构传送数据规则

为了叙述方便，体系结构的上一层数据(相应的客体记为 o)，经过封装后得到本层的新数据(相应的客体记为 o′)，以下简称客体 o 到达体系结构本层得到客体 o′。

·客体 o 到达体系结构第二层规则：

体系结构第一层客体 o 到达体系结构第二层，新客体为 apdu(o)，其规则定义为：

$travel_2$(apdu(o)，o：T(o)，I(o)，OR(o)，o_sa，o_rg，o fg)//表示拥有相关性质的客体 o 到达体系结构的第二层，得到客体 apdu(o)，有相应的性质。

If o fg⊄P fg then go end//对客体 o 进行细粒度控制
T(apdu(o))=T(o)，I(apdu(o))=I(o)，OR(apdu(o))=OR(o)，apdu(o) sa=o sa，apdu(o) rg=o rg，apdu(o)_fg=o_fg

·客体 o 到达体系结构第三层规则：

客体 apdu(o)到达体系结构第三层(安全插件层)后，记此时的客体为 P_i(apdu(o))，这里 P_i 表示不同的安全插件，定义规则为：

$travel_3$(P_i(apdu(o))，apdu(o)：T(apdu(o))，I(apdu(o))，OR(apdu(o))，apdu(o)_sa，apdu(o)_rg，apdu(o)_fg)//表示拥有相关性质的客体 apdu(o)到达体系结构的第三层，得到客体 P_i(apdu(o))，有相应的性质。

If apdu(o) sa⊄P then go end //若体系结构没有提供该安全服务，结束
T(P_i(apdu(o)))=P_i(T(apdu(o)))，
I(P_i(apdu(o)))=P_i(I(apdu(o)))，
OR(P_i(apdu(o)))=P_i(OR(apdu(o)))，
P_i(apdu(o)) sa=apdu(o)_sa，
P_i(apdu(o)) rg=apdu(o) rg，
P_i(apdu(o))_fg=apdu(o)_fg

规则说明：

本规则中：P_i(T(apdu(o)))，P_i(I(apdu(o)))，Pi(OR(apdu(o)))分别表示。客体 apdu(o)到达体系结构第三层(安全插件层)后，将根据在该层实施的不同安全规则 P_i 适当调整相应的密级、完整性等级和角色集，而不一定保持原值。具体怎样取值，可根据具体的安全插件来实施。具体情况另文描述。

·客体 o 到达体系结构第四层规则：

客体 P_i(apdu(o))到达体系结构第四层(会话连接控制层)后，记此时的客体为 S(P_i(apdu(o)))，定义规则为：

$travel_4$(S(P_i(apdu(o)))，P_i(apdu(o))：
T(P_i(apdu(o)))，I(P_i(apdu(o)))，OR(P_i(apdu(o)))，
P_i(apdu(o))_sa，P_i(apdu(o))_rg，P_i(apdu(o))_fg)

//表示拥有相关性质的客体 P_i(apdu(o))到达体系结构的第四层，得到客体 S(P_i(apdu(o)))，有相应的性质。

If o rg⊄P rg then go end//对客体 o 进行粗粒度控制
T(S(P_i(apdu(o))))=T(P_i(apdu(o)))，
I(S(P_i(apdu(o))))=I(P_i(apdu(o)))，
OR(S(P_i(apdu(o))))=OR(P_i(apdu(o)))，
S(P_i(apdu(o)))_sa=P_i(apdu(o))_sa，
S(P_i(apdu(o)))_rg=P_i(apdu(o))_rg，
S(P_i(apdu(o)))_fg=P_i(apdu(o))_fg

·经体系结构接收数据规则：

客体处于接收信息状态时，其信息流经过的过程与信息传送的过程刚好相反，所以客体经体系结构接收规则的描述，只要将客体经体系结构传送时的规则倒推。这里不再重复。

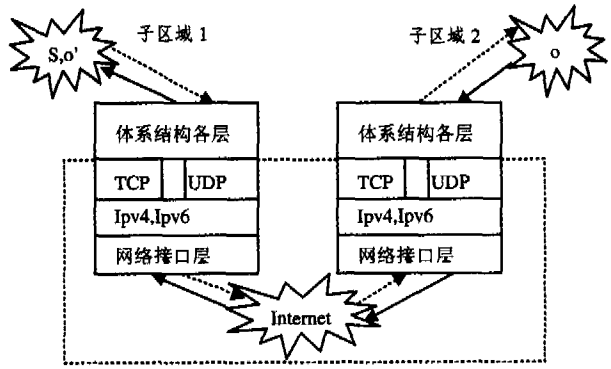

图 2 主体 s 与客体 o 的关系

3.3 其他规则

因篇幅所限，除上述最基本规则外，其他规则，不在此一一描述，读者可参阅文[6]、[7]、[9]。

3.4 规则的安全性分析

为了对上面描述的规则进行安全分析，设有一个主体 s 在网络的子区域 1 内，客体 o 处在子区域 2(如图 2)。这里子区域 1、2 假设是安全子域。

图中的矩形虚线围成的区域是体系结构外的部分，对它的形式描述，不是本文研究的范围。这里假设信息在此区域传输是安全的，且设没有发生信息流动时的系统初始状态是安全的。

子区域 1 中主体 s，欲对子区域 2 的客体 o 进行操作。下面对操作的安全性进行分析。

首先，子区域 1 中的主体 s 向网络发出申请(图中虚箭头)，请求对子域 2 中的客体 o 进行操作。当子域 2 得到相应申请后，将客体 o 通过体系结构和网络传输到子域 1 中，得到客体 o′，这里 o′实际上是客体 o 的备份。根据前面对体系结构中的安全规则的描述可知，只要体系结构的外部环境是安全的，信息从经过体系结构各层传送到经过体系结构各层被接收，不会影响其安全特性，即不会改变密级、完整性等级其他指标。这说明客体 o 及它的备份客体 o′具有相同的安全特性，从而主体 s 能对客体 o′进行安全操作，对客体 o 也能。反之，主体 s 能对客体 o 进行安全操作，对客体 o′也能。而本文所定义的规则，针对一般的安全子区域显然是安全规则。

由此，我们认为，本节给出的基于应用区域边界安全体系结构的描述规则是合理的、安全的，因此由这些规则组成的基于应用区域边界安全体系结构模型是安全的。

结束语 本文利用 BLP 模型、Biba 模型、RBAC 模型和信息流模型的基本性质，通过对文[5]提出的应用区域边界的安全体系结构进行分析，给出了该体系结构的描述规则。经对这些规则的分析，我们认为它们是合理的、安全的，相应的模型是安全的。通过对应用区域边界的安全体系结构的模型描述和验证，将有助于推动信息安全体系结构的理论研究。

参 考 文 献

1 GB/T 9287.2-1995. 信息处理系统．开放互连基本参考模型第 2 部分：安全体系结构

2 Kent S. Security Architecture for the Internet Protocol. RFC 2401, 1998.11

3 Information Assurance Technical Framework 3.1, 2002.9. http://www.iatf.net/framework_docs/version-3_1/index.cfm

4 沈昌祥. 构造积极防御的安全保障框架[J]. 计算机安全，2003，10：1～2

5 陈兴蜀. 应用区域边界的安全体系结构及实用模型研究[M]：[学位论文]. 成都：四川大学，2004

6 Bell D E, Lapadula LJ. Secure compter system [R]: mathematical foundation. MTR-2527, Mitrecorp, Bedford, MA, 1973 (NTIS AD771543)

7 Biba K. Integrity Considerations for Secure Computing Systems [R]: [Mitre Report MTR-3153]. Mitre Corporation, Bedford, MA, 1975

8 Sandhu RS, Samarati P. Access control: principles and practice [J]. IEEE communications. 1994,32(9): 40～48

9 刘益和. B/S 模式信息安全系统的一种形式化描述[J]. 计算机科学，2004,31(9A)：217～219

(上接第 69 页)

当 *str*1 小于 *str*2 时，返回负值。当 *str*1 等于 *str*2 时，返回 0。当 *str*1 大于 *str*2 时，返回正值。在使用时应该包含文件 string.h 库函数。

(2)函数-ftime

函数原型 void -ftime(strrct -timeb * timeptr) timeptr 指向 SYS\TIME.H 中定义的结构指针。

函数-ftime 读取当前时间并将其存放到由指针 timeptr 指向的结构中。-timeb 结构在 SYS\TIMEB.H 中定义。函数-ftime 的四个域及其取值如表 1 所示。

表 1 函数-ftime 的四个域及其取值

域	取值
dsfflag	如果当地正采用夏令时，此域值是非零(参见-tzset 中对定义夏令时的解释)。
millitm	不到一秒部分的毫秒数。最后一位数字总是 0，因为 millitm 每次增加近 1%秒。
time	从 1899 年 12 月 31 日午夜(00:00:00)计算起的秒数。
timezone	通用协调时间向西与当地时间的差，以秒为单位 timezone 的值是根据全局变量-timezone(参见-tzset)的值设定的。

-ftime 函数为 timeptr 所指向的结构域赋值。它不返回值。使用它时应该包含库函数 #include〈sys\type.h〉或 #include〈sys\timeb.h〉。

(3)getline()函数

函数原型为 istream&::getline(char * pszTarget, int nCount, char delim='\n')。

实现的功能：从文件中读取一整行字文本包括空白字符。参数 pszTarget 是用于存放读取的文本的字符数组，参数 nCount 是最多读取的字符个数，delim 是作为读取结束标准的分隔符。默认的分隔符是'\n'，但是不将分隔符存入缓冲区使用时应该包含库函数 #include〈fstream.h〉。

(4)函数 strlen()

函数原型 unsigned int strlen(char 8*str*)。

函数的功能统计 *str* 中字符的个数(不包括终结符'\0'，返回字符的个数。应包括 #include〈string.h〉。

6.5 时间和空间复杂度分析

算法的执行时间分为两个部分，散列计算的时间(记为 $T_{计算}$)和散列查找的时间(记为 $T_{查找}$)两个部分，所以算法的执行时间为 $T_{计算}+T_{查找}$。通常计算的时间要远小于访存的时间，也就是查找的时间。所以算法的时间约等于 $T_{查找}$，而散列查找的时间 T_s 要取决于冲突的次数。当查找的关键字是 N，散列表的基本区的大小是 M 的时候，散列查找的平均次数是 $N/2M$，所以假设流的数目是 N，散列表的基本区大小是 M，散列查找的平均次数是 $N/2M$，所以算法的时间复杂度是 $O(N/M)$。同时我们也指出，算法中使用的流的局部性原理可以加速查找过程，最好的情况下一次查找就能得到结果。

算法的占用存储空间主要是散列表的存储空间 U_{Hash}。而散列表主要用于对系统中的流进行存储。假设系统中流的数目是 N，且每个流的记录要占用 K 个字节，则 U_{Hash} 要占用的存储空间是 $K\times N$，即 $O(N)$。所以算法所占用的存储空间大约是 $O(N)$。

结论 程序的运行结果根据不同的仿真环境，结果有所不同。影响结果的因素有，仿真的硬件环境(即仿真计算机的配置)，所确定的散列表的容量，处理的数据包数，处理数据包的相关性，包在散列表中的活跃时间，定时扫描的时间等。本算法中，得到的结果是 0.0003 秒。即每 0.0003 秒处理一个数据包，每秒处理 3333 个数据包，每个数据包的平均长度是 36bit，则数据流量是 120kbit/s。

参 考 文 献

1 Algorithms for Packet Classification. http://itpapers.zdnet.com

2 Packet Classification Repository. http://www.ial.ucsd.edu/classification/

3 Telikepalli A. 数据包处理方法和解决方案[J]. 今日电子，2002，7：21～25

4 小高知宏. TCP/IP 数据包分析程序篇[M]. 北京：科学出版社，2003

5 Comer D E, 等著. 张娟，等译. 用 TCP/IP 进行网际互联(第二卷：设计、实现与内核)[M]. 北京：电子工业出版社，2003

6 Flow classification. http://www.hifn.com/technology/Classification.html