一种面向应用系统的强制访问控制模型

徐 璐,张红旗,杜学绘,王 超
XU Lu,ZHANG Hong-qi,DU Xue-hui,WANG Chao

解放军信息工程大学 电子技术学院,郑州 450004
Institute of Electronic Technology,The PLA Information Engineering University,Zhengzhou 450004,China
E-mail:xulu506@sina.com

XU Lu,ZHANG Hong-qi,DU Xue-hui,et al.Application system-oriented mandatory access control model.Computer Engineering and Applications,2010,46(9):107-110.

Abstract: It is not applicable to implement BLP in application systems.So an Application System-Oriented Mandatory Access Control(ASOMAC) model extended from BLP is proposed to fit the characteristics and requirement of application systems,which combines the application of role and security label to implement principle of least privilege and separation of duty,and the agility of mandatory access control is improved.The formal definition and theorem system are presented.Finally the design and analysis of mandatory access control system based on ASOMAC are performed.

Key words: mandatory access control;application system;security label;role

摘 要:为解决传统 BLP 模型不适用于应用系统的问题,提出了一种面向应用系统的强制访问控制(ASOMAC)模型,该模型对 BLP 模型进行了扩展,通过角色与安全标记的结合运用,实现了最小权限和职责分离原则,有效提高了强制访问控制的灵活性,使其符合应用系统的特点和需求。对模型进行了形式化定义,给出了一套公理系统,最后设计并分析了基于该模型的强制访问控制系统。

关键词:强制访问控制;应用系统;安全标记;角色
DOI:10.3778/j.issn.1002-8331.2010.09.031 **文章编号**:1002-8331(2010)09-0107-04 **文献标识码**:A **中图分类号**:TP309

随着各种网络威胁不断增多,应用系统对安全的需求越来越迫切,人们逐渐意识到安全而高效的访问控制技术对于复杂应用系统的重要性。目前国内外对于应用系统访问控制的研究很多,但大多集中在自主访问控制及基于角色的访问控制[1-3]等方面,而具有更高安全性的强制访问控制技术则较多地被应用在安全操作系统中。

但是对于操作系统来说,有时并不能有效获取真实的执行主体,比如在 FTP 服务中,操作系统在其访问控制监控上得到的执行主体是启动 FTP 的管理员, 而不是提出访问请求的用户,对 FTP 资源的访问控制只能依靠 FTP 应用程序来完成。因此对于安全需求较高的应用领域,如政府机关、军队等。目前如何在应用系统中实施强制访问控制是信息安全领域中亟待解决的关键性问题,建立灵活、高效的强制访问控制模型,是当前研究的难点。在 Bell-La Padula(BLP)模型的基础上进行了扩展,将角色与安全标记结合使用,构建了一种面向应用系统的强制访问控制模型。

1 强制访问控制模型

强制访问控制(MAC)[4]模型的基本思想是对访问的主体和客体均指定一个安全属性,主体对客体能否执行特定操作取决于二者安全属性之间的关系。

BLP 模型是应用最为广泛的经典强制访问控制模型,它通过安全级的偏序关系实现信息流的单向流动,以保证安全性和机密性,其一般规则有

(1)简单安全特性:仅当主体安全级支配客体安全级时,主体可以读客体。

(2)*-特性:仅当主体安全级被客体安全级支配时,主体可以写客体。

2 面向应用系统的强制访问控制模型

为提高应用系统安全性,达到安全标记保护级[5]的要求,需要在应用系统中实现强制访问控制。BLP 模型作为经典的多级访问控制模型,较多的被应用于安全操作系统,具有安全性较高的特点。但是,BLP 模型在实际使用中实现的工作量较大,不够灵活方便,而且过于强制保密性,对系统的连续工作能力,授权的可管理性方面考虑不足。而对于应用系统来说,往往需要在考虑安全性的同时,兼顾访问控制在使用和管理上的方便性和灵活性,并且对于完整性也有要求,此外,BLP 模型所定义的“只读”、“读写”、“只写”和“执行”4 种访问属性并不满足应用系统的需求。所以说传统的 BLP 模型不适合直接用于应用系

基金项目:公安部“金盾工程”(No.JIGAB23WD13)。

作者简介:徐璐(1984-),硕士研究生,主要研究领域为网络信息安全;张红旗(1962-),男,硕导,教授,主要研究领域为网络信息安全;杜学绘(1968-),女,硕导,副教授,主要研究领域为网络信息安全;王超(1975-),男,博士研究生,讲师,主要研究领域为网络信息安全。

收稿日期:2008-09-26 **修回日期**:2008-12-19

统的访问控制。

文献[6]提出了通过配置 RBAC 构造强制访问控制的方法，但这种构造方法对于用户来说较为复杂，若应用到实际中并不像使用一般的 RBAC 系统那样简单。文献[7]提出了一种融合了角色机制的强制访问控制模型，提高了 BLP 模型的灵活性，但该模型不是针对应用系统提出的，在访问属性、角色约束等方面不符合应用系统的特点和具体需求，也没有给出模型的公理系统，且并未实际应用于访问控制系统设计中。文献[8]在 BLP 模型的基础上，提出了一种扩展模型即面向应用的多级访问控制模型 AO-BLP，该模型扩充了 BLP 模型的访问属性，限制了 BLP 模型的下读和上写的范围，但并没有考虑最小和职责分离原则，易造成系统管理员和用户权限过大，而威胁应用系统的安全，此外该模型不支持新增加的访问属性，不符合实际应用的要求。

2.1 模型的建立

提出了一种面向应用系统的强制访问控制模型，在 BLP模型的基础上修改了元素的定义，并引入了角色集合，管理员集合，访问属性组集合，会话集合等元素，定义了映射函数及角色约束关系，同时对 BLP 模型的公理进行了改造。ASOMAC 的模型结构如图 1 所示。

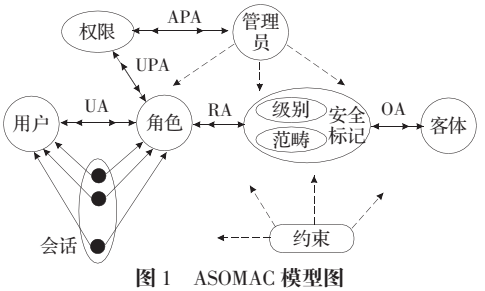

图 1 ASOMAC 模型图

用户是应用系统的使用者。角色是组织中的工作职能，用于描述分配给角色的用户所拥有的权限与职责。客体是应用系统中包含或接受信息的被动实体。访问属性是对客体的操作，需要根据应用系统的需求进行扩展。权限是角色可以对客体实施的已授权的相互作用的类型描述。会话是访问控制的一个单元，每个会话是动态产生的，与一个用户和该用户的角色相关联，一个用户可以同时创建多个会话。安全标记由级别和范畴组成，级别代表角色或客体的敏感程度，范畴代表角色或客体所隶属的范围，范畴一般依照单位和部门来划分，与具体的应用领域和信息特征有关。管理员独立于用户，在模型中执行管理功能。ASOMAC 模型的安全策略包括自主安全策略和强制安全策略。

对于作为特权用户的管理员来说，如果不对其拥有的特权进行限制，容易造成管理权限的失控，在这里将管理员的所有特权细分为若干特权子集合，并将这些特权分配给不同的管理员。在 ASOMAC 模型中主要有 3 类管理员：系统管理员，负责用户管理和角色管理，包括创建用户、删除用户、创建角色和删除角色；安全管理员，负责安全标记的指定以及安全策略维护，同时负责为用户指派角色并指定角色之间的约束关系；审计管理员，负责审计记录的管理与操作，分析审计记录并将分析结果向系统管理员反馈。每一个管理员都不能拥有所有的特权，并且没有访问客体的权限，只起到管理作用。由于实现了安全管理的三权分立，提高了安全性，避免了管理权限的滥用。

对于用户来说，系统安全策略往往要求尽量减少一个用户同时分配两个角色后产生欺骗行为的可能，例如一个用户如果同时被分配会计角色和出纳角色，就有可能产生欺诈行为。职责分离原则就是为了保证不出现这种错误，在 ASOMAC 模型中，通过定义与角色有关的约束来实现职责分离原则，在这里主要是指角色互斥和前提约束：

(1)角色互斥，同一用户仅可分配到互斥角色集合中至多一个角色，即静态互斥；或者用户虽然分配到互斥角色集合中的多个角色，但是在同一时刻用户至多能激活其中的某一个角色，即动态互斥。

(2)前提约束，可以分配角色给用户的前提条件是该用户已拥有另一角色。

2.2 形式化定义

下面给出了 ASOMAC 模型的形式化定义，具体描述模型中各个元素之间的关系以及模型的安全特性。

2.2.1 元素定义

(1)U，用户集合，$U=\{u_1,u_2,\cdots,u_n\}$。

(2)R，角色集合，$R=\{r_1,r_2,\cdots,r_s\}$。

(3)O，客体集合，$O=\{o_1,o_2,\cdots,o_m\}$。

(4)AR，管理员集合。

(5)E，互斥角色对集合，$E=SE\cup DE$，E 中任意元素记为(r_i,r_j)，且有 $SE\cap DE=\Phi$，其中 SE 为静态互斥角色对集合，DE 为动态互斥角色对集合。

(6)A，访问属性集合，如只读、读写、添加、执行、查询、复制、覆盖、删除、终止等，即 $A=\{\underline{r},\underline{w},\underline{a},\underline{e},\underline{l},\underline{c},\underline{o},\underline{d},\underline{s},\cdots\}$，访问属性集合中的元素可在实际应用中根据需要进行修改。

(7)AG，访问属性组集合，$AG=\{RG,WG,NG,EG\}$，RG 表示只读访问属性组，WG 表示读写访问属性组，NG 表示只写访问属性组，EG 表示执行访问属性组。各访问属性组由若干访问属性组成，如 $RG=\{\underline{r},\underline{l},\cdots\}$，$WG=\{\underline{w},\underline{c},\underline{o},\underline{d},\cdots\}$，其中只有 $NG=\{\underline{a}\}$，且不能被修改。所有的访问属性均需指定其所属的访问属性组，且只能属于一个访问属性组，管理员修改访问属性集合时，需同时对相关的访问属性组进行修改，即满足 $RG\cup WG\cup NG\cup EG=A$ 且各访问属性组集合两两交集均为 Φ。

(8)P，权限集合，分为用户权限集合和管理员权限集合，即 $P=UP\cup AP$。

(9)S，会话集合，$S=\{s_1,s_2,\cdots,s_i\}$。

(10)C，级别集合，$C=\{c_1,c_2,\cdots,c_q\}$，$c_1>c_2>\cdots>c_q$。

(11)K，范畴集合，$K=\{k_1,k_2,\cdots,k_t\}$。

(12)L，安全标记集合，$L=\{l_1,l_2,\cdots,l_p\}$，其中 $l_i=(c_i,K_i)$，$c_j\in C$ 且 $k_i\subset K$，这里定义 $l_1\geqslant l_2$ iff $c_1\geqslant c_2$ 且 $K_1\supseteq K_2$；$l_1\leqslant l_2$ iff $c_1\leqslant c_2$ 且 $K_1\subseteq K_2$；$l_1=l_2$ iff $c_1=c_2$ 且 $K_1=K_2$。

(13)D，判定集合，$D=\{\text{yes},\text{no},\text{error},?\}$。

(14)UA，用户集合到角色集合的多对多指派关系，$UA\subseteq U\times R$。

(15)PA，权限-角色的多对多关系，$PA\subseteq P\times R$。

(16)RA，角色-安全标记的多对一分配关系，$RA\subseteq R\times L$。

(17)OA，客体-安全标记的多对一分配关系，$OA\subseteq O\times L$。

(18)$b\subseteq(R\times O\times A)$表示在某个特定状态下，哪些角色以何种访问属性访问哪些客体。

(19)H，当前客体的层次结构，$o_j\in H(o)$表示在此层次结构中 o_j 为叶子节点，o 为父节点。

2.2.2 函数定义

(1)$role:U\rightarrow R$，将用户映射到其拥有的角色集合，$role(u)$，

$(u,role(u))\in U\times R$。

$crole:S\times U\to R$为用户 u 在当前会话 s 中激活的角色,$arole:U\to R$ 为用户 u 当前激活的角色集合,有 $crole(s,u)\in arole(u)$。

(2)$user:S\to U$,将会话 s 映射到其对应的用户 $user(s)$。

(3)F,安全标记函数,$F\subseteq L^R\times L^O\times L^C$,任意元素记为 $f=(f_R,f_O,f_C)$,$f_R:R\to L$, 是将角色映射到其拥有的安全标记的函数,$f_O:O\to L$,是将客体映射到其拥有的安全标记的函数,$f_C:U\to L$,是用于得到用户 u 在会话 s 中当前安全标记的函数,且有 $f_C(u)\leqslant f_R(crole(s,u))$。

(4)$g:R\times O\to R\times O\times A$,为角色权限函数。若 $g(r,o)=\Phi$,表示角色 r 对客体 o 不具有访问权限,不能对其进行访问;若 $g(r,o)=\{(r,o,\underline{x}_1),(r,o,\underline{x}_2),\cdots,(r,o,\underline{x}_n)\}$,表示角色 r 对客体 o 可进行的操作集合为$(\underline{x}_1,\underline{x}_2,\cdots,\underline{x}_n)$;若对于 $\forall r\in R$,均有 $g(r,o)=\Phi$,则表示该客体已经死亡。模型中用角色权限函数 g 代替 BLP 模型中的访问控制矩阵进行自主访问控制。

2.2.3　角色约束关系

(1)静态互斥,对于 $\forall r_i,r_j\in R$ 且 $i\neq j$,

$((r_i,r_j)\in SE)\wedge(r_i\in role(u))\Rightarrow r_j\notin role(u)$

(2)动态互斥,对于 $\forall r_i,r_j\in R$ 且 $i\neq j$,

$(((r_i,r_j)\in DE)\wedge(r_i,r_j\in role(u))\wedge(r_i\in arole(u)))\Rightarrow r_j\notin arole(u)$

(3)前提约束,对于 $\forall r_i,r_j\in R$ 且 $i\neq j$,

$r_i\in role(u)\Rightarrow r_j\in role(u)$

2.2.4　重要公理

(1)简单安全特性:状态 $v=(b,g,f,H)$满足简单安全特性 iff 所有的 $r\in R$,$o\in b(r:\underline{x}_1,\underline{x}_2,\cdots,\underline{x}_n)$有

$(\underline{x}_1,\underline{x}_2,\cdots,\underline{x}_n\in RG\cup WG)\Rightarrow(f_R(r)\geqslant f_O(o))$

其中 $b(r:\underline{x}_1,\underline{x}_2,\cdots,\underline{x}_n)$表示角色 r 对其具有访问属性 $\underline{x}_i(1\leqslant i\leqslant n)$的所有客体的集合。

(2)*-特性:状态 $v=(b,g,f,H)$满足严格的 *-特性 iff 对于 $\forall s\in S,r=crole(s,user(s)),o\in b(r:\underline{x}_1,\underline{x}_2,\cdots,\underline{x}_n)\Rightarrow$

$$\begin{cases}(\underline{x}_1,\underline{x}_2,\cdots,\underline{x}_n\in NG)\Rightarrow(f_C(user(s))\leqslant f_O(o))\\(\underline{x}_1,\underline{x}_2,\cdots,\underline{x}_n\in WG)\Rightarrow(f_C(user(s))=f_O(o))\\(\underline{x}_1,\underline{x}_2,\cdots,\underline{x}_n\in RG)\Rightarrow(f_C(user(s))\geqslant f_O(o))\end{cases}。$$

这里只允许对高等级数据区中的空白区域进行写入操作,避免了对高等级数据的盲写所带来的对信息完整性的破坏,满足了应用系统对完整性的要求。

(3)自主安全特性:状态 $v=(b,g,f,H)$满足自主安全特性 iff 对所有的$(r_i,o_j,\underline{x})\in b\Rightarrow(r_i,o_j,\underline{x})\in g(r_i,o_j)$。

(4)兼容性公理:状态 $v=(b,g,f,H)$满足兼容性 iff 对所有的 $o\in O$,有 $o_1\in H(o)\Rightarrow f_O(o_1)\geqslant f_O(o)$。

3　基于 ASOMAC 模型的访问控制系统设计

参照“安全标记保护级”的要求,基于 ASOMAC 模型,对某小型办公自动化系统的访问控制系统进行了设计。系统分为安全管理系统及安全控制系统,安全管理系统提供系统安全元素的管理和配置,以独立的系统形式提供给应用系统,而安全控制系统完成用户登录及用户访问控制裁决等功能。

强制访问控制根据安全标记来控制主体对客体的访问,因此需要在系统中保存角色与客体的安全标记信息。将用户与角色的相关信息存放在用户信息数据库中,包括角色分配与角色的安全标记信息等,由系统管理员和安全管理员管理。客体的安全标记信息存放在客体标记数据库中,由安全管理员管理,对于现有的客体,只要在标记数据库中为其建立对应的数据项,而不需要修改现有客体的结构和内容,就可以将它们纳入强制访问控制的管理之下,易于扩展现有的应用系统。同时,将安全标记集中存放同时也便于对其进行集中管理。图 2 所示为基于 ASOMAC 模型的访问控制系统设计。

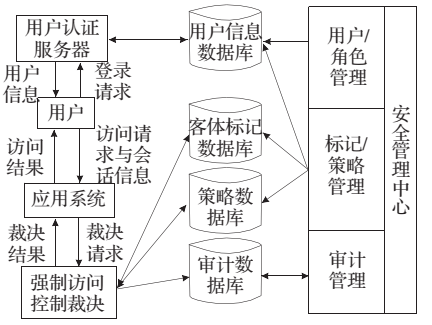

图 2　基于 ASOMAC 模型的访问控制系统图

3.1　访问控制流程

假设管理员已对系统进行了初始设置,考虑用户 u_i 对客体 o_j 进行访问,具体的访问控制流程如下:

(1)用户 u_i 首先通过认证服务器进行单点登录,通过认证后可得到其用户信息 $user.mes(u_i)$,用户信息包括用户拥有的角色集合 $arole(u_i)$及角色的安全标记,该用户信息由认证服务器进行签名,带有时间及频度限制。

(2)u_i 激活特定角色启动会话 s_k 时,生成当前会话信息 $session.mes(s_k)$,会话信息包括用户当前安全标记 $f_C(u_i)$、当前会话激活角色 $crole(s_k,u_i)$。传统的强制访问控制系统要求用户在登录系统时提供用户当前安全标记,如需更改当前安全标记,则需要退出系统重新登录,而在本系统中用户更改当前安全标记仅需重新启动新的会话。

(3)u_i 向应用系统提交访问请求 $request(u_i,s_k,o_j,\underline{x})$,并同时提交 $session.mes(s_k)$。

(4)应用系统收到访问请求后,分解出请求中的用户、客体及操作类型,并调用本系统提供的 API 函数,将这些信息及会话信息按照系统定义的通信协议格式封装后发送给访问控制裁决。

(5)访问控制裁决在收到用户的访问请求后,首先判断用户所提交的信息是否真实有效,如无效则直接返回失败信息。

(6)访问控制裁决执行访问控制裁决算法,将结果返回给应用系统。应用系统再调用系统提供的 API 函数对结果进行解析。如判定结果 $d\in\{yes\}$,则返回用户所请求的信息,如判定结果 $d\in\{no,error,?\}$,则返回失败信息,拒绝访问请求。

(7)裁决结束后,将审计数据存入审计数据库中,审计数据库由审计管理员进行管理。

3.2　访问控制裁决算法

访问控制裁决收到应用系统发送的裁决请求后,分解出传入的参数,即$(AppType,session.mes(s_k),u_i,r_i,o_j,\underline{x})$,其中 $AppType$ 为应用系统标识,$session.mes(s_k)$为会话信息,u_i 为发起会话的用户,r_i 为 u_i 在 s_k 中激活的角色,即 $crole(u_i,s_k)$,o_j 为客体,$\underline{x}$为操作类型。访问控制裁决根据 AppType 定位到策略数据库中该应用系统的安全策略数据区,之后执行访问控制裁决算法:

(1)查找白名单库,若有$(r_i,o_j,\underline{x})$项转向(5)。

(2)查找黑名单库,若有$(r_i,o_j,\underline{x})$项转向(6)。

(3)查找策略数据库,若$(r_i,o_j,\underline{x})\notin g(r_i,o_j)$则转向(6)。

(4)从 $session.mes(s_k)$中提取用户当前安全标记,并根据 o_j 的唯一标识符在客体标记数据库中查找 o_j 对应的安全标记,根据相应的规则比较用户当前安全标记与要访问的客体的安全标记以裁决用户的访问请求是否被允许。若允许访问转向(5),若不允许则转向(6),若有多个规则适用于这一请求则转向(7),若规则无法处理此请求则转向(8)。

(5)裁决结束,返回判定结果 d=yes。

(6)裁决结束,返回判定结果 d=no。

(7)裁决结束,返回判定结果 d=error。

(8)裁决结束,返回判定结果 d=?。

4 模型比较

通过对上述访问控制系统中角色授权、角色分配、策略配置及访问控制裁决等环节中性能及安全性的测试分析表明,ASOMAC 模型根据强制访问控制规则对用户访问请求进行严格的控制,具有较高的安全性。同时,在实际应用中,角色一般在数量上远远少于用户,且相比于用户较为固定,所以通过引入角色极大地简化了为用户分配权限时的工作量,提高了访问控制的可管理性。表 1 所示为 ASOMAC 模型与其他几种模型的比较。

表 1 ASOMAC 模型与 DAC、BLP、RBAC 模型的比较表

	ASOMAC	DAC	BLP	RBAC
安全性	高	低	高	中
可管理性	高	低	低	高
复杂度	高	低	高	高

5 结论

通过分析应用系统访问控制的特点,针对应用系统的需求提出了一种面向应用系统的强制访问控制模型,该模型在继承 BLP 模型安全性优势的同时对其进行了改进与扩展,借鉴基于角色的访问控制思想,提高了强制访问控制在应用系统中的适用性,并在实践中得到了应用。

参考文献:

[1] Sandhu R S,Coyne E J,Feinstein H L.Role-based access control models[J].IEEE Computer,1996,29(2):38-47.

[2] Ferraiolo D,Barkley J.A role-based access control model and reference implementation within a corporate intranet[J].ACM Transactions on Information and System Security,1999,2(1):34-64.

[3] Joshi J B D,Bertino E,Latif U,et a1.A generalized temporal role based access control model[J].IEEE Trans on Knowledge and Data Engineering,2005,17(1):4-23.

[4] Bell D E,Lapadula L J.Secure computer system:Unified exposition and multics interpretation,MTR-2997 Rev.1[R].The Mitre Corporation Technical Report,1976.

[5] 中华人民共和国质量技术监督局.GB17859-1999 计算机信息系统安全保护等级划分准则[S].1999.

[6] Osbom S,Sandhu R,Munawer Q.Configuring role-based access control to enforce mandatory and discretionary access control policies[J].ACM Transactions on Information and System Security,2000,3(2):85-106.

[7] 陈四清,王家耀,李波,等.融合角色机制的强制访问控制模型[J].计算机工程与设计,2007,28(24):5870-5873.

[8] 陈伍军.面向应用的访问控制研究[D].南京:南京大学,2004.

(上接 87 页)

3 结论及展望

通过对 TCP Veno 在 TCP Reno 协议的基础上修改后进行分析,在仿真的 MANET 环境下用 NS2 软件对 TCP Veno 的测试,得出了一些对 TCP Veno 理论研究和应用有一定参考价值的测试结论:在存在背景、有随机丢包、并且存在拥塞的MANET网络中,Veno 的性能优于 Reno,而且在背景流越大,达到拥塞的时间越短、随机丢包越大,Veno 的优越性更会非常明显。同时还得出随着跳数的增加、链路误码率的增加,TCP 的吞吐量显著下降;虽然 TCP Veno 的性能在高误码率的 MANET 下较TCP Reno 好,但是随着跳数的增加,这种优势并不明显。因此 TCP Veno 较适合于跳数较少的无线链路,在这样的链路中,即使误码率较高,TCP Veno 也能较好地保持 TCP 的吞吐量性能。对 TCP Veno 在 MANET 环境下的测试,还有很多问题是可以继续研究的,如何解决 Veno 在 MANET 环境中比较高的重传率;如何解决 Reno 和 Veno 两种协议的兼容性问题,都可以作为进一步的研究方向。

参考文献:

[1] Fu Cheng-peng.A remedy for performance degradation of TCP vegas in asymmetric networks[EB/OL].(2003-12-20)[2008-09-10].http://www.ie.cuhk.edu.hk/fileadmin/s.

[2] Chung Ling Chi,Fu Cheng-peng,Liew Soung Chang.Improvements achieved by SACK employing TCP Veno equilibrium-Oriented mechanism over lossy networks[EB/OL].(2001-12-12)[2008-09-10].http://www.ie.cuhk.edu.hk/index.php?i.

[3] Fu Cheng-peng.TCP Veno:TCP enhancement for transmission over wireless access networks[EB/OL].(2003-11-20)[2008-09-10].http://scholar.ilib.cn/Abstract.aspx.

[4] Liu J,Singh S.ATCP:TCP for mobile ad hoc networks[J].IEEE Journal,2001,19(7):1300-1315.

[5] Paxson V,Allman M.Computing TCP's retransmission timer[EB/OL].(2000-11-15)[2008-09-10].http://www.cis.umassd.edu/~vvokkarane/courses/cis577/f08/papers/rfc2988.

[6] Allman M,Balakrishnan H,Floyd S.Enhancing TCP's loss recovery using limited transmit[EB/OL].(2006-09-20)[2008-09-10].http://www.cnpaf.net/class/rfcen/06111141842601821 26.html.

[7] Handley M,Padhye J,Floyd S.TCP congestion window validation[EB/OL].(2000-06-12)[2008-09-10].http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.1493.

[8] Zou Zi-xuan,Lee Bu Sung,Fu Cheng-peng.Packet loss and congestion state in TCP VENO[C]//Proceedings 12th IEEE International Conference on Network,2004,2:16-19,731-735.

[9] Altman E,Jimenez T.Novel delayed ACK techniques for improving TCP performance in multihop wireless networks[C]//Proc of the Personal Wireless Communications,Vemce,Italy,2003:237-253.

[10] 欧阳志鹏,沈富可.一种无线 Ad hoc 网络拥塞的解决方案[J].微型电脑应用,2006,22(2).

[11] 彭海英,蔚承英,唐红.TCP Veno 在 3G 环境下的性能测试[J].计算机工程与设计,2007,28(18).

[12] Chen K,Xue Y,Shah S,et al.Understanding bandwidth-delay product in mobile ad hoc networks[J].Elsevier Computer Communications,2004,27:923-934.