

Jabber 无线环境可靠传输协议的设计与实现

杨 云, 张 洁

(陕西科技大学 电气与信息工程学院, 陕西 西安 710021)

摘 要: 提出采用 UDP 协议, 在 C 语言编程环境下设计和实现基于 UDP 协议的 Jabber 系统, 从而达到最大限度发挥无线网络传输协议本身功能的目的. 并在 Jabber 系统模块中, 实现无线网络中的可靠传输和连接保持等功能.

关键词: Jabber; 无线网络; TCP; UDP; 可靠性

中图分类号: TN915

文献标识码: A

文章编号: 1000-7180(2010)06-0134-03

Design and Implementation of Reliable Transfer Protocol of Jabber in Wireless Environment

YANG Yun, ZHANG Jie

(Department of Electrical and Information Engineering, Shaanxi University of Science & Technology, Xi'an 710021, China)

Abstract: This paper adopts UDP to design and implement a realization of Jabber in C language programming environment, and achieves several main functions in Jabber models, such as reliable transfer and connection keeping in wireless network.

Key words: Jabber; wireless network; TCP; UDP; reliability

1 引言

无线网络中基于 IP 分组的传输层协议除 TCP 协议外还包括 UDP 协议^[1]. 基于 XMPP 的 Jabber 系统目前所采用的传输层协议为 TCP 协议. 在无线网络环境中, XMPP 协议的数据描述语言 XML 虽然功能强大, 但同时也占用了较多的网络资源. 另外, TCP 长连接的方式无论从信道资源消耗还是计算资源消耗来讲, 都不太适合 Jabber 系统在无线网络环境下的应用. 而 UDP 协议具有简单、无连接、实时性高、信道资源占用率低等适应无线网络环境的优点, 被大部分通信软件采用; 它除了在协议报头中定义了 4 个字段外, 该协议没有说明任何命令. UDP 报头用来说明消息的源和目标端口, 除了使用一个字段来说明数据报的长度外, 还给出了一个用来表示整个数据报加上前置报头的数值校验和字段, 建立在封装的 IP 报头基础上的前置报头的逻辑结构

为消息的路由选择和报头纠错提供了 UDP 报头结构、用户接口和 IP 接口^[2]. 这些特点使它与无线网络的数据量小、发送时间间隔大、不定时发送等相匹配, 更好的适应了无线网络的通信环境.

为了发挥 UDP 协议的优点、弥补无线网络中 TCP 协议的不足, 文中提出在无线网络环境中采用 UDP 协议, 利用 C 语言编程在 Jabber 系统架构中通过增加应用层 ARQ 重发机制、心跳机制和超时逐出机制实现可靠传输和连接保持等技术.

2 Jabber 系统下信息的可靠传输算法的设计与实现

由于 UDP 具有不可靠传输的特点, 在 Jabber 中通过在应用层增加 ARQ 重发机制来保证数据报的顺序处理; 针对通信双方中任何一方发生网络中断, 另一方无法感知的情况, 通过增加心跳机制和超时逐出机制来维持通信双方的连接状态.

收稿日期: 2009-05-15; 修回日期: 2009-08-15

基金项目: 国家自然科学基金重点项目(59732050); 陕西省自然科学基金基础研究计划项目(SJ08E103)

2.1 应用层 ARQ 重发机制

UDP 是一种不可靠的传输,物理链路错误、网络设备故障、路由错误、网络拥塞都有可能造成 UDP 报文的丢失,而且协议本身不提供对数据报丢包后的重发功能.考虑到系统 I/O 复用的特殊模型,应用层应该设计一个合理的重发机制,是 I/O 复用的补充.对于按顺序发送的数据报,接收端收到数据报的顺序可能是乱序,因为网络拥塞、路由状态变化、路由表更新等都会导致数据报从不同的路由到达目的接收端,传输时延也各不相同.为保证发送的报文能够按顺序被处理,应用层的重发机制也应该保证数据报能够按照顺序进行处理.基于以上几点,在此实现了应用层 ARQ^[2]重发机制,该机制能有效地对发送报文进行管理,并可以保证 UDP 报文的可靠传输、顺序处理.

应用层 ARQ 重发机制算法的实现:

(1) 增加 XML 报文报头信息.该报头尽可能短,以减少网络流量.报头由 2 个字节组成,共 16bit.对于已发送报文的应答请求,没有正文部分,仅有报头 2 个字节的确认信息.报头结构如图 1 所示.

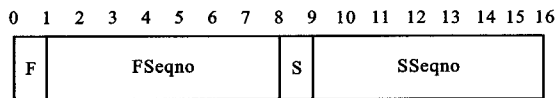


图 1 增加 XML 报文报头结构

该报头的第 1bit 表示数据报的发送方,设置一个布尔型变量 F 为标志位.当服务器发出报文以及客户端对该报文应答时,该位置为 1;当客户端发出报文以及服务器对该报文的应答该位置为 0.

第 2 到第 8 位为发送报文的序号 SN 以及应答报文的序号 RN,合称为 FSeqno.服务器端启动时将报文 SN 初始化为 0,客户端登陆后将 RN 初始化为 0,即 SN 表示发送的报文编号,RN 表示对 SN 的确认以及请求的下一条报文的编号.报文编号采用 mod128 编号,编号范围为 0~127.

第 9 位表示报文是否为已经拆分过的报文编号 S.如果是拆分过的报文置位 1,如果不是拆分过的报文或者是拆分的报文中的最后一条,则置为 0.

第 10 位到第 16 位为拆包的小编号 SN1 和应答请求 RN1,合称为 SSeqno.表示这是一个长报文拆分后的第几个报文,或者对第几个报文的应答以及下一条拆包报文的请求,该编号也采用 mod128 编号,编号范围为 0~127.报头信息中不包含 XML 正文长度信息,由 UDP 协议自己校验;这里需要说

明的是,服务器端对于每一个 session,都维持独立的一套 SN-RN 编号;客户端仅仅维持一套 SN-RN 编号.

(2) 发送方发出报文时,调用 get_head(sess_t sess, char * new_data, const char * old_data, int fflag, int sflag) 函数为发送的报文添加头部信息,然后发送出去,并将发出报文保存在 session 结构体发送缓冲区成员 s_data * data_send 中,同时增加时间戳;接收方通过 mio 回调函数收到的报文,不直接交给 sx 回调函数进行处理,而是首先通过函数_del_head(sess_t sess, char * data)函数将数据报报头去掉,然后将之保存在 session 结构体接收缓冲区成员 r_data * data_recv 中.

(3) mio 回调函数接收到数据之后,首先通过_data_check 函数判断该报文是对已发出报文的应答还是对端主动发出的报文.如果是应答请求,则根据应答的 RN 编号,找到对应的已发送报文,将报文交给 sx 回调函数处理,并将已确认的发送报文删除;如果是对端主动发来的报文,则将该报文存入 session 接收缓冲区,并发送对该报文的确认,流程如图 2 所示.

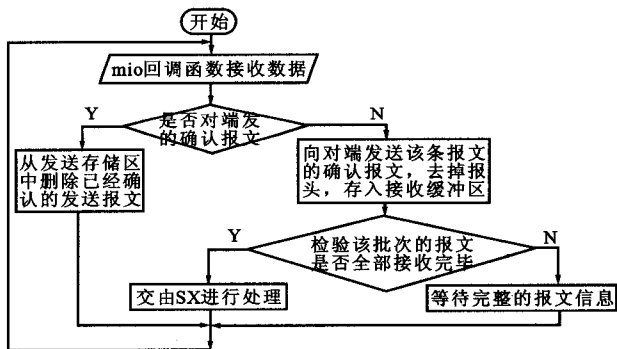


图 2 报文处理流程图

(4) c2s 主进程执行完 select 操作,完成接收到达的报文之后,则遍历二级 session 哈希表,依次检测所有 UDP^[3]的 session 的发送缓存区 data_send,根据每条报文的发送时间戳,判断是否重发,超时可定为 2 秒.可以允许每条报文重发 15 次,如果重发 15 次后还是没有收到对该消息的确认,则认为该用户已经非正常登出系统,将该用户踢出服务器.

2.2 心搏机制

由于 UDP 是一个无连接的协议^[4],通信的双方中任何一方发生网络中断,另一方是无法感知的.因此通过增加心搏机制,可以维持通信双方的连接状态.服务器端和客户端的对心搏机制的实现基本一致,都是在 I/O 复用之后完成.

心搏机制的算法实现:

为 session 设置一个连接保持时间戳成员,当某个 session 有数据报到达时,即有读事件发生时,则更新该时间戳成员为当前时间.

时间戳的更新仅以读事件为准,发送报文时,不更新时间戳.

当 session 与通信对端没有发生任何交互时,设置一个 io_check_keepalive 值,一旦没有任何为网络 I/O 事件发生的时间超过该值,则向对端发送一个字节的数数据报,报文长度为一个字节,内容为一个空格字符.

接收端 session 接收到一个字节的报文,并且报文内容为一个空格字符时,仅仅更新该 session 的时间戳,不做其他处理.

心搏机制的实现,是在 I/O 复用 select 之后执行.首先遍历 UDP 用户的 session 二级哈希表,找到每一个 session 的读事件的时间戳,一旦该值大于 io_check_keepalive 后,则立刻执行发送一个空格报文的的操作,这是对服务器而言,对客户端而言,直接检查 session 时间戳,超时则发空格报文.

2.3 超时逐出机制

服务器端和客户端均设置超时逐出机制.因为 UDP 协议是无连接的协议,一旦通信的一端网络发生故障,另一方需要通过一定的手段,使自己确认同通信对端已经无法连接.服务器端和客户端的超时逐出机制实现基本相同,不同的是服务器端需要对所有用户的 session 进行超时逐出判断.

2.3.1 超时逐出机制算法

(1) 无论是客户端还是服务器端,每一个 session 均对读事件设置时间戳.对于网络正常连接的判断,均以时间戳是否超时为准.需要说明的是 2.2 节中的心搏机制主要是为了在通信的双方没有任何 xml 流交换的情况下,主动发起的读事件驱动,目的是为了更新连接保持时间戳,作为网络连接的判断依据,超时逐出机制实际是对心搏机制无法正常维持的实际操作.

(2) 服务器端在 select 的 I/O 复用结束之后,立刻开始遍历所有 UDP 的 session,以读事件的时间戳为判断依据.一旦检测时间与时间戳的差值超过一个限制,如 65 秒,则调用 sx_close 函数结束服务器端与该客户端的 xml 流交换,通过发送 <stream:stream> 分节,结束这个 session,对该用户进行踢出操作.这里需要注意的是服务器端也要遍历 UDP 的 session 二级哈希表.对于客户端的登出,由客户端自

己实现超时逐出机制.

(3) 客户端也是在 select 的 I/O 复用之后,可是检测与服务器端保持通信的 session,一旦检测时间与时间戳的差值超过一个限制,本系统中设定为 65 秒,则调用 sx_close 函数结束该客户端与服务器端的 xml 流交换,再做登出操作.对于服务器端,该用户的登出通过服务器端的超时逐出机制完成.

2.3.2 超时逐出机制的算法实现

```
while(1)
{
    mio_udp_run(c2s->mio,5);
    while(jqueue_size(c2s->dead_sess)>0)
    {
        sess=(sess_t)jqueue_pull(c2s->dead_sess);
        if(sess->ip!=NULL)free(sess->ip);
        if(sess->result!=NULL)nad_free(sess->result);
        if(sess->jid!=NULL)jid_free(sess->jid);
        if(sess->fd->sess_list!=NULL)jid_free(sess->jid);
        {
            back_unique(unique_tag,sess->tag);
            sess_list_de(sess->fd,sess);
            sess_want_write_list_de(sess->fd,sess);
        }
        free(sess);
    }
    While(jqueue_size(c2s->dead)>0)
        sx_free((sx_t)jqueue_pull(c2s->dead));
    if(c2s->io_check_interval>0 && time(NULL)>=c2s->next_check)
    {
        c2s_time_checks(c2s);
        c2s->next_check=time(NULL)+c2s->io_check_interval;
    }
}
```

3 结束语

对 UDP 协议加以改进,使其在实际应用中可以根据使用环境实现 TCP 协议和 UDP 协议的可选^[5-6],达到最大范围内发挥自身的优势,在网络传输过程中实现传输效率更高,接收误码率更低,通信质量更好,网络可靠性更高的效果.由于用户所能感知到的网络服务的性能主要取决于网络中通信协议的性能,所以网络通信协议的改进也能增加网络使用的用户数目.

(下转第 141 页)

(9) 逻辑规则是由谓词和动作以及逻辑连接符和时态操作符组成的. RBUCON 中的逻辑规则的BNF 语法定义如下:

$$\phi ::= a \mid p(t_1, t_2, \dots, t_n) \mid (\neg \phi) \mid (\phi \wedge \phi) \mid \forall x : \phi \mid \exists x : \phi \mid \square \phi \mid \diamond \phi \mid \bigcirc \phi \mid \phi \cup \phi \mid \blacksquare \phi \mid \blacklozenge \phi \mid \phi \Theta \mid \phi S \phi$$

其中, a 表示动作, p 表示 n 元谓词, $t_1, t_2, \dots, t_n$ 是谓词的项, x 是变量. $\neg, \wedge, \rightarrow$ 是逻辑连接符, 其语义分别是“非”, “与”, “蕴涵”; $\square, \diamond, \bigcirc, \cup, \blacksquare, \blacklozenge, \Theta, S$ 是时态操作符, 其语义分别是“Always”, “Eventually”, “Next”, “Until”, “Has - always - been”, “Once”, “Previous”, “Since”.

通过以上的 TLA 逻辑分析可知, 在用户访问资源的整个过程中, 使用控制决策要受到授权规则谓词、条件谓词和属性更新动作的限制, 上述因素的变化导致角色状态的变化(撤销、中止、挂起和恢复等), 从而支持了动态授权. 它使资源的使用始终处于合法受控的状态, 相对于静态授权, 能提供更全面的访问控制, 避免静态授权造成权限泄露的产生, 从而提供了更高的安全方式, 在很大程度上满足了实际应用需要和最小特权原则.

5 结束语

文中综合 RBAC 和 UCON 模型的优势提出了一种基于角色的使用控制授权 RBUCON 模型. 在使用定义其基本元素上的基础上, 采用 TLA 逻辑对 RBUCON 动态授权过程进行了分析. 在 RBUCON 模型中, 基于一系列授权规则 AR 来进行角色分配, 角色权限的激活要受到上下文环境的约束, 简化了

授权管理, 更好地支持动态授权, 增强了授权管理的灵活性和安全性. 下一步工作是探讨支持 RBUCON 模型在 PMI 中的应用和动态访问控制策略.

参考文献:

- [1] 芮国荣, 邢桂芬. 基于角色和规则的访问控制[J]. 计算机应用, 2005, 25(4): 864 - 866.
- [2] 姚寒冰, 胡和平, 李瑞轩. 上下文感知的动态访问控制模型[J]. 计算机工程与科学, 2007, 29(5): 1 - 3.
- [3] 崔永泉, 洪帆. 基于使用控制和上下文的动态网络访问控制模型研究[J]. 计算机科学, 2008, 35(2): 37 - 41.
- [4] Ravi Sandhu, Jaehong Park. Usage control: a vision for next generation access control [C]// Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies. New York: ACM Press, 2004: 57 - 64.
- [5] 桂劲松, 陈志刚, 邓晓衡. 一种改进 UCONc 的服务网格授权决策模型[J]. 西安电子科技大学学报, 2008, 35(3): 546 - 553.
- [6] Zhang X, Park J, Parisi - Presicce F, et al. A logical specification for usage control [C]// Proc. of the 9th ACM Symposium on Access Control Models and Technologies. New York, 2004.

作者简介:

谢辉 男, (1983 -), 硕士研究生. 研究方向为授权管理与访问控制.

张斌 男, (1969 -), 博士, 副教授, 硕士生导师. 研究方向为信息安全.

张红旗 男, (1962 -), 教授, 硕士生导师. 研究方向为信息安全.

单宝恒 男, (1983 -), 硕士研究生. 研究方向为授权管理与访问控制.

(上接第 136 页)

参考文献:

- [1] Rind Abdul Razaque, Shahzad Khurram, Qadir M Abdul. Evaluation and comparison of TCP and UDP over wired - cum - wireless LAN [C]// 10th IEEE International Multi - topic Conference 2006, INMIC. Islamabad, Pakistan, 2006: 337 - 342.
- [2] 李建东. 信息网络理论基础 [M]. 西安: 西安电子科技大学出版社, 2002.
- [3] 方乐. 基于无线局域网的 UDP 及 TCP 性能分析 [D]. 上海: 复旦大学, 2000.
- [4] 王毅. 使用 UDP 协议实现数据可靠传输 [D]. 北京: 北京科技大学, 2005.

- [5] Degermark Mikael, Engan Mathias, Nordgren Bjorn, et al. Low - loss TCP/IP header compression for wireless networks [J]. Wireless Networks, 1997, 5(3): 375 - 387.

- [6] Rosenberg J. A framework for conferencing with the STP [EB/OL]. (2006 - 02 - 13) [2009 - 03 - 01]. <http://www.ietf.org/rfc/rfc4353>.

作者简介:

杨云 女, 教授, 硕士生导师. 研究方向为通信网络计算机应用、图形图像处理.

张洁 女, 硕士研究生. 研究方向为 GSM 网络优化的数据分析.