第 36 卷 第 9 期 计 算 机 工 程 2010 年 5 月
Vol.36 No.9 Computer Engineering May 2010
·开发研究与设计技术· 文章编号：1000—3428(2010)09—0279—03 文献标识码：A 中图分类号：N945

嵌入式 iSCSI 协议的简化与实现

贺再红，张 艳
(湖南大学计算机与通信学院，长沙 410082)

摘 要：研究 iSCSI 协议的嵌入式固件简化方法，提出一种无需操作系统支持的嵌入式 iSCSI 通信协议原型，建立一个嵌入式 iSCSI 启动器有限状态模型，该模型便于操作系统内核 iSCSI 启动器的平滑接管。将简化协议集成到 IP-SAN 扩展 BIOS 中，并应用于无本地存储设备的主机上，实现基于 IP-SAN 的网络引导。

关键词：嵌入式 iSCSI 协议；状态模型；远程引导

Simplification and Implementation of Embedded iSCSI Protocol

HE Zai-hong, ZHANG Yan
(School of Computer and Communication, Hunan University, Changsha 410082)

【Abstract】The embedded firmware simplification method for iSCSI protocol is researched. An embedded iSCSI communication protocol prototype without Operating System(OS) support is proposed. The embedded iSCSI initiator finite state machine model is set up, which is easy for smoothly taking over the OS kernel mode initiator. The simplified protocol is integrated into IP-SAN extension BIOS, and applied to client host with no local storage devices. It achieves network boot based on IP-SAN.

【Key words】embedded iSCSI protocol; state model; remote boot

1 概述

iSCSI 协议[1]基于成熟的 TCP/IP 协议及广泛使用的以太网，将 SCSI 协议映射到 TCP/IP 协议上，使 SCSI 的命令、数据和状态可以在传统的 IP 网络上传输，充分利用现有技术和设施，非常适合用来构建低成本的 IP-SAN 系统，为中小企业提供了低投入、高性能的网络存储解决方案。

基于 iSCSI 的 IP-SAN 远程引导技术支持客户主机开机后经授权认证自动连接到 iSCSI SAN 上，并将 SAN 上的存储卷映射给主机，可以从逻辑硬盘上引导启动系统软件与应用软件。用户根据需要从服务器下载、运行所需要的操作系统和应用程序，然后在客户机本地进行计算，从而极大地降低客户机系统的成本，简化管理和维护，同时增强系统的安全性。

针对如何将分配给某个登录用户的存储卷映射成该远程用户主机的本地逻辑硬盘问题，本文设计并实现了一种嵌入式 iSCSI 启动器，以便将远程用户的 I/O 访问重定向到网络存储卷。

2 嵌入式iSCSI协议原型

嵌入式 iSCSI 启动器由于没有操作系统的支持，可供利用的软件平台只有客户主机 BIOS 提供的软中断服务，在 CPU 工作于实模式的情况下，可以使用的存储器资源非常有限，所以，该 iSCSI 启动器代码要求短小，且分配的数据缓冲区也要尽可能少。

为了解决资源限制问题，需要根据嵌入式 iSCSI 启动器运行特征，对 iSCSI 网络协议做原型系统设计以及功能需求裁减。

本文基于 RFC3720 兼容性规范提出一种可移植的嵌入式协议实现原型，如表 1 所示。它取消或简化了一些需求，设计了新的状态转换机制，同时保证了 iSCSI 的性能。

表 1 固件协议功能需求原型

RFC3720 功能需求	嵌入式协议栈裁减/支持	说明
iSCSI 协议序列机制	部分支持	支持大部分命令，支持立即数据，不支持 NOPIN, NOPOUT 状态序列
iSCSI 协议会话机制	支持	单一会话并建立单一连接，只支持常规型会话
iSCSI 登录和协商机制	部分支持	选择使用部分参数
错误处理和恢复机制	支持	支持级别 0，即 Session 层的错误恢复
iSCSI 状态转换机制	支持 5 个状态	归并 3 个状态，新增 FINISHED 状态以便实现模式驱动到保护模式内核驱动的平滑过渡
iSCSI 其他机制	部分支持	不支持 AHS,SNACK PDU，IPV6 寻址，重定向，任务管理，消息同步和数据导航机制等

3 嵌入式iSCSI启动器简化与实现

3.1 嵌入式iSCSI启动器状态模型

由于所需构建的嵌入式 iSCSI 启动器基于可利用资源非常有限的裸机平台，且只需要在引导远程操作系统时运行，发挥作用的时间相对较短，并具有被接管的特性，因此并不需要实现所有的标准启动器状态表和所有状态转换。

标准 iSCSI 启动器连接(connection)状态模型如图 1 所示，其中，状态集 Q 和状态转移条件集 Σ 如下：

Q={$S1$, $S2$, $S4$, $S5$, $S6$, $S7$, $S8$}($S3$ 和 $T3$ 仅对目标器有效)

Σ={$T1$, $T2$, $T4$, $T5$, $T7$, $T8$, $T9$, $T10$, $T11$, $T12$, $T13$, $T14$, $T15$, $T16$,

基金项目：教育部博士点基金资助项目(200805321029)；湖南省自然科学基金资助项目(07JJ6139)

作者简介：贺再红(1972－)，女，讲师、硕士，主研方向：网络存储，嵌入式系统及应用；张 艳，讲师、硕士

收稿日期：2009-12-12 **E-mail：**hezaihong168@yahoo.com.cn

*T*17, *T*18}

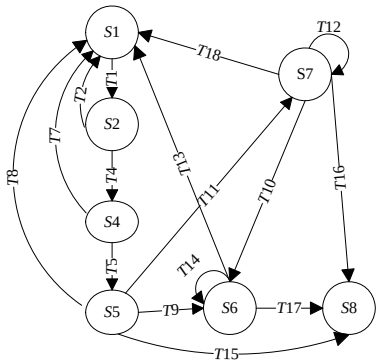

图 1　标准启动器连接状态模型

从图 1 可以看到，标准 iSCSI 启动器连接状态数较多，且不同状态间的转换比较繁琐，不太适合嵌入式固件应用环境的要求。通过对状态节点集合分别进行划分和合并，以达到减少节点个数的目的[2]。基于嵌入式固件系统的特殊应用，本文提出以下状态合并替换方法：

状态 *Si* 要替换 *Sj*，如果满足以下 3 个条件：(1)状态 *Sj* 的任意状态跳转分支，*Si* 基本能提供；(2)*Si* 不会扩展 *Sj* 的操作，即 *Sj* 的发送或接收消息操作基本都会出现在 *Si* 中；(3)*Si* 终止，*Sj* 也终止，则 *Si* 可以替换 *Sj*。

对于标准 iSCSI 启动器连接状态集 *Q* 中的 *S*6, *S*7, *S*8 分别对应退出登录中、等待退出登录(logout)和等待清除状态，对于这 3 个状态，考虑到嵌入式启动器需要简洁性、使用时效短和易于接管特性，直接用一个状态 *Sn* 归并。对于 *T*9, *T*10, *T*11, *T*12, *T*15, *T*16 等状态变迁，因为嵌入式启动器基于单连接单任务，所以可以将过程简化为：当启动器收到异步消息时，启动器退出登录(logout)当前的连接，释放清理当前会话(session)；当启动器向目标器发送了退出登录后，关闭当前会话；当 TCP 连接被复位，关闭或连接超时，并恢复当前会话。简化后的状态集 *Q* 和状态转移条件集*Σ*如下：

Q={*S*1, *S*2, *S*4, *S*5, *Sn*}

Σ={*T*1, *T*2, *T*4, *T*5, *T*7, *T*8, *Tn*}

图 2 为简化后的嵌入式 iSCSI 启动器状态模型。

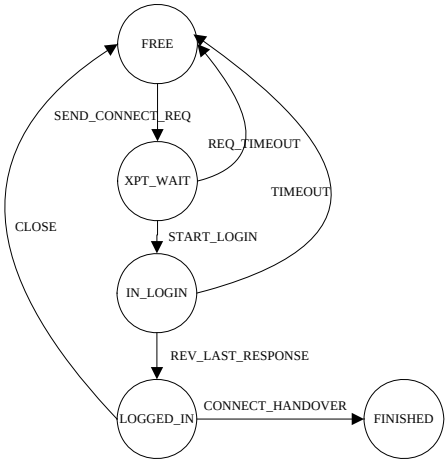

图 2　嵌入式启动器状态转换模型

在建立连接之前，启动器处于 FREE(*S*1：连接初始化或成功关闭连接时的状态)状态。当启动器发送建立连接请求后，转移到 XPT-WAIT(*S*2：发送 TCP 连接请求，等待建立连接请求的响应)状态，等待目标器的响应。此时如果连接超时，仍然没有接收到目标器的响应，启动器将回到 FREE 状态重新发送请求。目标器接收到该连接请求后等待登录(login)开始。启动器向目标器发送登录请求后，转移到 IN-LOGIN(*S*4：发送初始登录请求，等待最终登录响应)状态，等待目标器的响应。

在成功登录之后，启动器转移到 LOGGED-IN(*S*5：LOGIN 过程结束，处于全功能阶段)状态，此时进入全功能数据收发阶段。FINISHED(*Sn*：连接被操作系统内核的 iSCSI 启动器接管)为归并后的自定义状态，处于该状态时，嵌入式 iSCSI 启动器被操作系统内核 iSCSI 启动器成功接管，相应的目标器端清理此会话连接，嵌入式 iSCSI 启动器被终结。

3.2　嵌入式iSCSI启动器会话状态

iSCSI 连接在相应的 iSCSI 会话中创建，如图 3 所示，在会话没有建立之前，启动器处在 FREE(会话尚未建立或成功清除后的状态)状态。当 iSCSI 连接进入 LOGGED-IN(等待任何一种会话事件的状态)状态时，启动器的状态转移到 LOGGED-IN 状态，如果此时会话失败，那么启动器的状态将转移到 FAILED(等待会话恢复或会话被终止的状态)状态来进行会话恢复，恢复完毕后状态将重新转移到 LOGGED-IN，会话操作继续，当嵌入式 iSCSI 启动器被接管后，启动器会话被终止、清理。

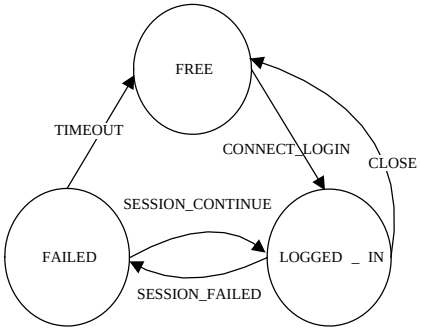

图 3　嵌入式启动器会话状态

通常一个 iSCSI 启动器会话在 FFP 阶段，会创建 2 种独立线程：Tx 发送线程和 Rx 接收线程，用于发送和接收 iSCSI PDU。它们分别守候在相应的命令队列的同步对象 Semephore 上，当新的命令插入或到达队列，Tx 或 Rx 被唤醒，把所有应该发送的命令或响应发送出去后，然后重新在 Semephore 上等待[3]。

在嵌入式 iSCSI 启动器实现中，没有相应的信号量等资源可用，因此，将相应的发送线程与接收线程简化成单任务，在会话和连接的实现过程中，直接使用一系列函数调用来实现 iSCSI 连接有限状态机，负责整个事务处理的建立请求、响应、结果和确认。

iSCSI 会话函数被初始化后，将调用连接实现函数来构建 iSCSI 连接有限状态机，成功建立起和 iSCSI 目标器的连接后，这时嵌入式 iSCSI 核心就可以全面进行命令、数据传送和执行。当嵌入式 iSCSI 启动器会话处理完毕后，会话(session)将发送结束命令给连接(connect)，让其结束和关闭事务处理。

3.3　错误处理和恢复

由于高级别的错误恢复能力需要更复杂的系统设计和更多的系统资源支持。显然这不符合嵌入式 iSCSI 协议原型简洁高效的要求，在可供利用资源非常有限的情况下，只选择

支持会话层次的错误恢复。嵌入式 iSCSI 启动器真正发挥作用的时间并不是很长，主要是在系统启动的初始阶段，在将服务器端的操作系统文件加载后就被接管。在发起连接时，启动器端首先置好 ErrorRecoveryLevel 值为 0，然后向目标器端发出协商请求，建立连接。

在会话交互过程中，如果目标器方发现某连接出现错误，它会向启动器发出“异步消息”，通知启动器中止该连接。在启动器收到目标器的消息和响应后，会发起会话层次的错误恢复。会话恢复意味着连接的关闭，在目标器端终止掉所有与指定启动器相关的正在执行和正在等待的任务，在启动器端用一个适当的 SCSI 服务应答码终止掉所有输出的 SCSI 任务，并且重新建立一个包含新的连接集合的会话。包括启动器向目标器重新发起 TCP 连接、登录等过程。

4 测试与评价

基于以上嵌入式 iSCSI 协议原型与设计方法，本文在 Intel 82546EB 双端口千兆网卡和 Relteak 8139 百兆网卡的扩展 ROM 上开发了 IP-SAN 扩展 BIOS，实现了无本地存储设备的主机远程引导，测试环境如下：

(1)存储系统平台

Red Hat Linux64 SL4.2/NSS(包括 iSCSI Target V3.0+ Storage Volume Management System V3.0+DHCP Server V3.0)/Intel Xeon 3.6 GHz×2/2 GB Mem/ RAID/Intel 双千兆网卡；客户主机平台：AMD AthlonXP 2.0 GHz/1 GB Mem/带嵌入式 iSCSI 扩展 BIOS 的 Intel 82546EB 双端口千兆网卡或 RealTek 8139 百兆网卡；网络连接：1 000 Mb/100 Mb 交换以太网。

(2)测试任务

基于嵌入式 iSCSI 扩展 BIOS 远程引导过程路径失败“愈合”性能、与同类对比。

在系统远程引导过程中，拔掉连接客户机和存储系统之间的网线，造成路径失败，以考察连接循环等待的时间和性能，随后再重新接入网线，以考察路径性能。

当连接失效时，基于嵌入式的固件协议将进入连接的循环等待恢复连接阶段，如果在拔掉网线后 3 min 内重新接入网线，连接将恢复，引导过程仍然可以继续，保证了连接的可靠性。该远程引导方式与 PXE 引导方式的比较结果如表 2 所示。

表 2　iSCSI-BIOS 启动测试及对比结果

	访问方式	启动平台	映射硬盘大小	直接安装操作系统和应用软件	扩展性	直接分区和格式化
iSCSI Boot BIOS	磁盘设备块	任何操作系统，包括 Dos,Windows (Win98/Me,2k/XP 等)及 Linux	可为任意大	可以直接安装多个操作系统	可动态扩容而不会破坏已有数据	可以
PXE Boot	文件	Dos,Windows98, Linux，不能启动 Win2K,WinXP	只能为 2 GB	不能	容量固定，不能动态扩容	不能

测试结果表明，基于嵌入式 iSCSI 映射出的 SCSI 硬盘与客户端直接安装的本地硬盘呈现出一样的引导性能。与 PXE 相比，后者是基于文件级的[4]，只能映射出一个大小受文件系统限制的启动盘，须先在客户端硬盘上安装操作系统和应用软件，然后再上传，配置较复杂。而前者是基于设备级的，可根据需要映射出多个任意大小的磁盘，可在任何一个磁盘上直接安装操作系统和应用软件。

5 结束语

本文提出一种结构清晰、占用系统资源少、可移植性强的嵌入式 iSCSI 原型系统，设计并实现了嵌入式 iSCSI 协议，将其嵌入到 PCI 网卡 ROM 内的 IP-SAN 扩展 BIOS，将裸机客户机端的网卡模拟成 SCSI 主机总线适配器，从而实现了 IP-SAN 存储卷的远程映射与引导，相比以往同类技术具有较大的优势。

参考文献

[1] Tom C. IP SANS: A Guide to iSCSI, iFCP, and FCIP Protocols for Storage Area Networks[M]. [S. l.]: Pearson Education, 2002.

[2] Abhijeet J. A Scalable and High Performance Software iSCSI Implementation[C]//Proc. of the 4th Int'l Conference on File and Storage Technologies. San Francisco, USA: [s. n.], 2005.

[3] 谭怀亮, 李仁发, 贺再红. 基于 iSCSI 网络存储协议的映射 SCSI 磁盘启动方法[J]. 计算机工程与应用, 2006, 42(18): 109-112.

[4] 李　蕾, 苏金树, 张银福. iSCSI 协议中错误恢复机制的设计与实现[J]. 计算机工程, 2006, 32(2): 119-121.

编辑　陈　文

（上接第 278 页）

```
                ATC_AGP,&pn_num);
for(;;)   {
          if( soft_status = 1){  /*应用进程为主状态，进行业务处理*/
          …
          }
          else if( soft_status = 2){
          /*应用进程为备状态，进行状态监视*/
          …
          }
        else if( soft_status == 0 )      {
  /*应用进程为中间态，等待状态的确定*/
          …
        } } }
```

5 结束语

针对空管系统现有的双机管理子系统，本文提出基于代理检测的应用级双机热备份设计方案及软件实现，该方案在原通用双机管理基础上进行改进，达到应用进程间的主备，实现负载均衡的功能。经过在国产首套空管自动化系统中的实际运行检验，证明该方案工作稳定可靠，状态转换更加平滑，完全可以满足空管系统中双机冗余的设计，也具有在其他空管自动化系统中推广的价值。

参考文献

[1] 黄　锴, 许志伟. 可扩展并行计算技术、结构与编程[M]. 北京: 机械工业出版社, 2000.

[2] 刘　东, 张春元, 李　瑞. 基于任务同步的双机容错系统[J]. 计算机工程, 2007, 33(8): 224-226.

[3] 王泽均, 陈　新, 王　勇. 基于动态负载均衡的网络监控系统[J]. 计算机工程, 2008, 34(24): 115-117.

编辑　陈　文