

基于 X.400 系列建议的电子邮政系统

张玉忠 (河南师范大学) 赵建宾 张德运 (西安交通大学)

摘要:本文介绍了一个基于 X.400 系列标准而设计并在微机上实现的电子邮政系统的总体结构及主要模块功能,并给出了用户代理 UA 及传输代理 MTA 的实现算法流程图。对 MHS 在微机上的实现作了研究和探讨。

1. 引言

世界上先进国家已十分普及使用的电子邮政系统(MHS),我国也开始使用和推广。要在我国推进 MHS 的应用,就需要在微机上开发相应的 MHS。随着高档微机的出现,性能价格比的提高,系统软件的丰富,使在微机上开发 MHS 已成为可能。本工作就是在此背景下,作为多功能网络(MFNET)开发工作的一部分提出来的。

2. 实现的目标及设想

MHS 提供了多种服务元素,在仔细分析现有条件下层所能提供的支持后,我们提出可以实施的现阶段目标:根据 X.400 系列建议,从功能上实现人间合作 UA 及传输系统所提供的基本服务,处理字符型正文,系统兼含 UA 和 MTA 功能,提供多用户服务,实现 P1 和 P2 协议。

由于 MFNET 下层的工作正在进行,为了能够检验所实现的功能,我们利用同台主机两个终端间的通信模拟下层提供的服务,进行文电交换。即 RTS 用户通过服务原语与 RTS 进行交互,使网络下层的改变不会影响到 MHS 的实现,以后下层开发完毕,只需对 RTS 进行必要的修改,就能投入运行。

MHS 软件的开发采用软件工程的规范,分阶段、分过程实现。首先,在对功能标准和系统开发环境进行分析的基础上,确定实现目标,提出系统结构和功能模块,并逐步细化。具体实现时,采用自顶向下的技术,分功能、分模块进行。

作为操作系统支持下的若干应用程序,对于 MHS 中不同层次的软件,其工作方式各有不同。UA 软件是系统和用户间的接口,应具有和用户交互的功能,所以它由用户键入命令予以启动,作为进程,在分时状态下运行,

并与 MTA 软件交互原语,向用户提供 MHS 服务;其它各层软件,则作为后台进程,由系统管理员提交运行,并以进程通信的方式实现各层之间的原语交互。

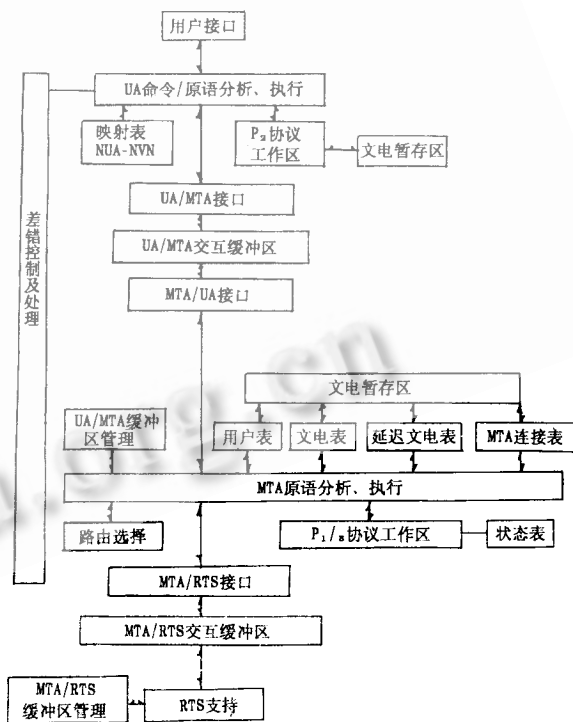


图1 MHS在微机上的总体结构

对接口的设计,由于不同接口所处位置不同,实现的方法各有差异。用户接口本着用户使用方便的原则,采用菜单方式进行工作,所提供的菜单选择项应包括所实现的全部 MTL 服务原语的操作;层间接口则根据对实

现系统支持能力的分析,采用消息传送作为通信方式,实现服务访问点(SAP)功能;内部接口采用过程调用和参数传递的方式予以实现。

本系统是在 386 微机及 XENIX 操作系统环境下实现的,主机有 12 个虚拟终端,通过多用户卡,还可外接 4 个物理终端,可供 16 个用户同时注册。

3. 系统总体结构

具备协议标准和实现环境的支持,就奠定了实现 MHS 的基础,通过对协议标准和系统环境的研究和分析,提出了 MHS 具体实现的系统总体结构,如图 1。

各功能框图的简要说明如下:

- (1) 用户接口:实现系统和用户的交互;
- (2) 层间接口:实现相邻层间的原语交互;
- (3) 原语/命令分析及执行模块:对接收到的原语或命令进行分析,并进行相应处理;
- (4) 协议工作区:根据不同层次,按对应协议处理文电;
- (5) 缓冲区:作为相邻层间原语交互的通道,采用消息队列实现;
- (6) 映射表:实现网络用户名(NUN)和网络用户地址(NUA)之间的映射;
- (7) 用户表:记录某一 MTA 中已注册的用户状况;
- (8) 文电表:记录某一 MTA 中所有等待处理的非延迟投递文电;
- (9) 延迟文电表:记录用户已提交,但投递时间未到延迟递交文电;
- (10) MTA 连接表:记录 MTA 的连接状况;
- (11) 路由选择:确定文电的传输途径;
- (12) 状态表:记录原语交互所带来的状态变化,确保顺序的正确;
- (13) 缓冲区管理:进行层间原语交互缓冲区的分配、回收等管理工作。

4. 系统的实现

(1) 层间接口的实现。按照 OSI 参考模式,层间通信通过服务访问点 SAP 进行,其实现方法依据实系统的支持环境和所选用的程序设计语言,常见的方法有过程调用、公共缓冲区、队列、文件共享等,本系统采用进程间通信的方法。

XENIX 操作系统提供了丰富的进程通信手段,基本的有信号、FIFO 和管道、高级的有消息传送、信号量和

共享存储器。通过分析比较,本系统主要以消息队列作为通道,交互原语。

根据消息传送的特性,一个消息队列可作为双向通道,实现一对进程间的通信。由于系统对消息队列数目有限制,必须采取一种合理算法,将有限的消息队列分配给众多的请求者。本系统采用先进先出的动态管理算法,将通道分为公用和专用两种,前者仅用于处理 UA 登录请求和 MTA 对此请求的响应,它由 MTA 启动后建立,并常驻系统,可被多个 UA 使用;后者用于其它原语的交互,一个专用通道在某一时刻只能分配给一个 UA 占用,仅当占有者释放后,才可提供给另一 UA 占用。任何 UA 在任何时候都可通过公用通道发送请求原语,而 MTA 根据先来先服务的原则管理专用通道。没有获得专用通道的 UA 无权与 MTA 交互其它原语。

(2) UA 软件的实现。UA 软件由用户启动运行,如用户地址正确,则形成一个 UA 进程提供有关服务;否则,显示用户地址非法,退出系统。其工作流程如图 2(a)所示;执行命令示例(登录)如图 2(b)所示。

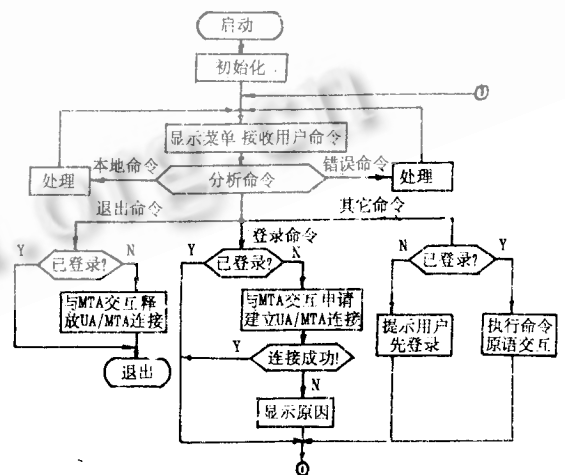


图 2 (a) UA 工作流程

(3) MTA 软件的实现。MTA 软件是 MHS 的核心,由系统管理人员启动,作为后台进程运行,与上下层交互原语,并定时检查有关文电表,处理文电。一旦 MTA 链接表上有等待发往外地的文电,则根据 RTS-连接的情

况作相应处理。如果连接未建立好,则通过有关原语,请求建立 RTS-连接;否则,在占有权标的条件下,利用已建立的连接传输文电。

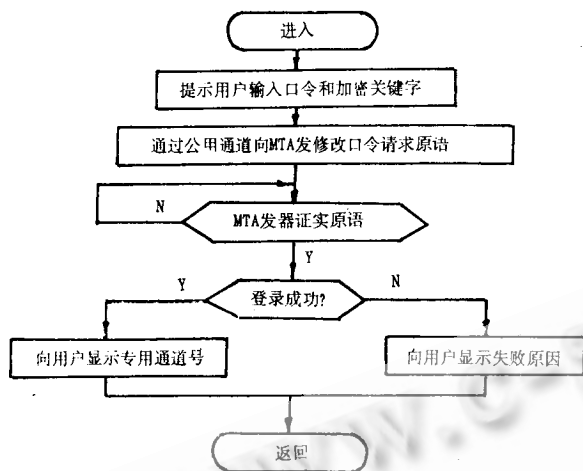


图2 (b) 执行命令示例(登录)

需要指出的是,我们把延迟投递服务在本地实现,这样处理尽管与延迟投递的原意不符,但有两个好处,一是便于延迟投递服务元素的实现,二是与其它独立开发的系统互连时,即使对方不提供延迟投递服务,本方仍具备此项功能。

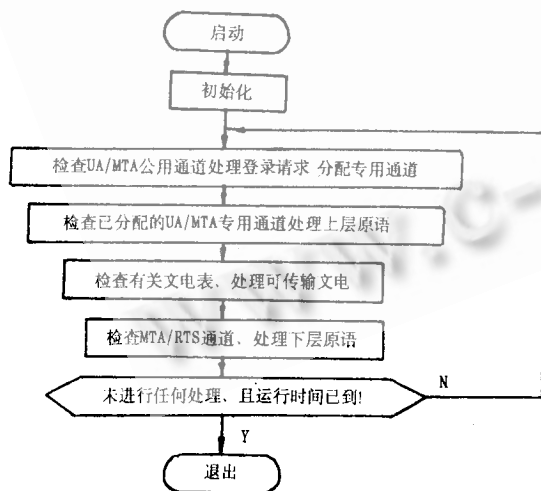


图3 (a) MTA 工作流程

MTA 工作流程如图 3(a)所示;处理用户登录请求的流程如图 3(b)所示。

(4)RTS 的模拟实现。RTS 作为 MHS 的一个单独子层,用一独立模块实现。它通过服务原语,向 MTA 提供服务。本系统所利用的 RTS 服务原语是按照 X.410 建议而定,并增加了 TRANSFER 的响应和证实原语,这主要考虑到 MTAE 需要了解所发送的 APDU 是否已正确传输,以便决定下一步的处理;同时,非证实原语无法让 MTAE 知道所发原语是否被下层接收及处理。

本系统属我们自主开发的多功能网络(MFNET)的应用层软件。由于下层的开发工作尚在进行,本系统无法直接利用下层的支持,所以只能采用模拟 RTS 所提供服务的办法来实现 MTAE 间 APDU 的透明传输。在这里,假设下层已建立起可靠连接,使用消息队列来传送 APDU。实际运行结果表明,这种方法是可行的。

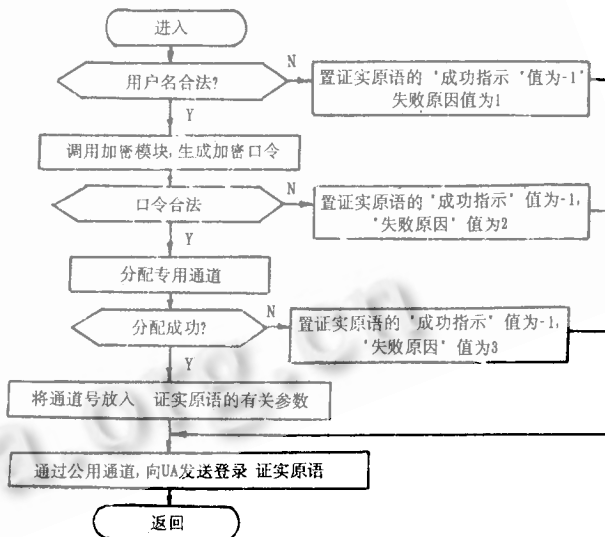


图3 (b) MTA 处理登录请求流程

参考文献:

- [1] CCITT, Recommendations X.400-X.430 (1984)
- [2] CCITT, Recommendations X.400-x.420 (1988)
- [3] 赵小凡: "信报处理系统(MHS)标准", 计算机应用与软件 第7卷第2期 1990年
- [4] 张德运编著: <<通信及网络理论基础>> 西安交大 1991年
- [5] 中国科学院软件所编译: <<XENIX System v 丛书>> 北京科学技术出版社 1990年